Learning multi-way associations across biological networks

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Zhuliu Li

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Rui Kuang

April, 2021

Acknowledgements

There are many individuals who have earned my gratitude for their contribution to my graduate school time at the University of Minnesota.

My heartfelt and deepest appreciation goes first and foremost to my advisor, Dr. Rui Kuang, for providing me the wonderful opportunity to study under his supervision. All his motivation, guidance, encouragement and patience during my journey as a graduate student were crucial for my growth and the creation of high-quality research products to complete my dissertation. Apart from his broad and solid knowledge of computational biology and machine learning, I was also greatly inspired by and benefited from his passion for science, critical thinking, and efficient and effective communication. I am immensely fortunate and proud that I had him as my advisor.

Furthermore, my sincere gratitude also goes to alumni: Dr. Wei Zhang, Dr. Huanan Zhang, Dr. Raphael Petegrosso, Dr. David Roe, and Dr. Catherine Lee, and members: Tianci Song, Nishitha Paidimukkala, Yao Gong, Kaige Zhang, Shrijana Gurung, Charles Broadbent and Sharada Sridhar in the Computational Biology Laboratory directed by Dr. Rui Kuang. I am grateful for their support and friendship during my personal and professional time in Minnesota. Their generosity with their time and ideas was also important to the success of my Ph.D. program.

In addition, I would like to thank the members of my thesis committee: Dr. Yousef Saad, Dr. Jun Sun and Dr. Jeongsik Yong for their time, attention, insightful questions and constructive feedback on my research work.

Finally, I am sincerely thankful to my family for their continuous and unparalleled love and support.

Dedication

To my parents, my wife and my daughter.

Abstract

The multi-way associations across multiple biological networks carry rich information of the dependencies between heterogeneous biological objects such as human diseases, genes (proteins) and drugs. Inferring this information based on the network topologies has emerged as a core step in many bioinformatics tasks. Due to the modeling and computational challenges, current studies mainly focus on predicting the bipartite associations across pairs of biological networks rather than predicting the multi-way associations; and most of the existing multi-relational learning methods are not scalable for predicting high-order associations across a large number of networks.

The goal of our research is to advance the field of multi-relational learning by systematically addressing the modeling and computational challenges. To that end, we develop machine learning methods to directly model the multi-way associations across multiple biological networks as a tensor (multi-way array), which is structured by the manifolds in the product of multiple networks; and improve the learning scalability through optimally estimating the tensor and the product graph. These methods rely on tensor decomposition and graph-based transductive learning technologies. First, we propose the Graph-Regularized Tensor Completion from Observed Pairwise Relations (GT-COPR) method to directly infer the multi-way associations among the entities across multiple biological networks in a low-rank tensor using the observed bipartite associations. We validate that compared to the state-of-the-art bipartite relational learning methods, the tensor formulation enables GT-COPR to identify more important pharmacogenomic multi-way associations across disease, gene and chemical networks. Next, we present the Low-Rank Tensor-based Label Propagation (LowrankTLP) method to address scalability challenges in multi-relational learning, by providing a theoretically justified framework to estimate the product graph with respective to the learning objective. The large-scale experiments demonstrate that LowrankTLP indeed well approximates the original learning objective with remarkably improved scalability and accuracy. Finally, we introduce the Fast Imputation of Spatially-Resolved Transcriptomes by Graph Regularized Tensor Completion (FIST) method, to impute the missing mRNA expressions in the spatial transcriptomics RNA sequencing (sptRNA-seq) data

in a compressed tensor. FIST is the first method to model the multi-way spatial and functional associations between genes and tissue locations as a tensor, and significantly improves the imputation accuracy by leveraging the tensor modeling of the sptRNA-seq data.

We comprehensively evaluate our methods and discoveries with simulations and real biological datasets. The results suggest that the tensor modeling of the multi-way data and the integration of topological information carried in multiple networks as the product graph, improve the quality and significance of the inferred multi-way associations using the proposed methods; the principled estimations to the tensor and product graph improve the scalability of these methods, and enable them to learn very high-order multi-way associations across a large number of networks for a variety of real applications.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Biological networks present processes in cells, organisms, or entire ecosystems [1]. A variety of biological networks have been constructed to represent the interactions between proteins, regulatory associations between transcription factors and genes, biochemical reactions between compounds, and signals that are transduced between cells, etc. As follows, we list the biological networks that will be analyzed in this thesis.

- *Protein-protein interaction (PPI) networks* are mathematical representations of the physical contacts between proteins in the cell, where the nodes are proteins and the edges indicate the interactions between proteins. There are diverse methods to identify the edges, such as the most commonly used yeast two-hybrid system (Y2H) experiment [2, 3].

- *Common disease network* [4] reveals the phenotypic overlap between common diseases (nodes) in human. To generate the edges in the network, each disease is first annotated with Human Phenotype Ontology (HPO) [5] terms which provide a standardized vocabulary of phenotypic abnormalities encountered in human disease. An edge between two diseases is drawn if the similarity between their phenotypic profiles exceeds a threshold.

- *Chemical similarity network* represents the structural connection between chemicals, where chemicals are nodes and their structural similarities are edges. Specifically, each chemical is represented using PubChem [6] 881-bit structure fingerprints [7], then two chemicals are connected by an edge if the Tanimoto coefficient [8] between their fingerprints is higher than a threshold.

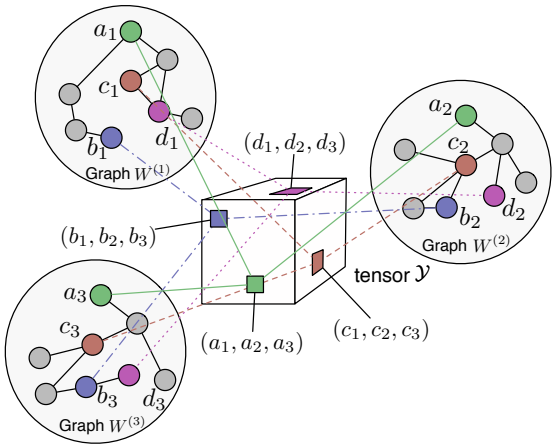


Figure 1.1: **Learning multi-way associations across biological networks.** Given n (illustrated by $n = 3$) different biological networks represented as undirected graphs with adjacency matrices $\{W^{(i)} : i = 1, \ldots, n\}$, multi-relational learning infers the associations across the nodes of n graphs in an n-way tensor $\mathcal{Y}$. Each tuple of the n nodes in the same color is represented as an entry in the tensor.

Many research topics in bioinformatics are driven by the analysis of the biological networks. One important topic is to learn the multi-way associations among the entities across multiple biological networks, which reveals the dependencies between heterogeneous types of biological objects, and characterizing their functional roles. For example, learning multi-way associations among diseases, genes and chemicals from content-rich biomedical and biological networks can provide important guidance for drug discovery, drug repositioning and disease treatment [9–11]; aligning the nodes of multiple protein-protein interaction (PPI) networks across different species is helpful for finding evolutionary conserved pathways, protein complexes and functional orthologs [12–15]; exploring the phenome-genome associations from the PPI network and the ontology

hierarchies of human genes and phenotypes can drive the discovery of novel molecular targets of cancers [16–18]; imputation of spatially-resolved transcriptomes considering the multi-way associations between genes and tissue locations can overcome high dropout rate of mRNAs in in-situ capture and complete the profiling of the gene expressions [19–21].

1.2 Multi-relational learning task

As the tensor (multi-way array) is a natural representation of the multi-way associations, the learning task introduced in Section 1.1 can be converted to a network-guided tensor completion problem. In this section, we mathematically define the multi-relational learning problem studied in this thesis. Let $W^{(i)}$, $D^{(i)}$ and $L^{(i)} = D^{(i)} - W^{(i)}$ be the adjacency, degree and graph Laplacian matrices of n different undirected graphs $\{G^{(i)} = (V^{(i)}, E^{(i)}) : i = 1, \ldots, n\}$, where $V^{(i)}$ and $E^{(i)}$ denote the node and edge sets of the i-th biological network. The multi-relational learning objective is to predict the n-way associations among the nodes across the n graphs in an n-way tensor $\mathcal{Y}$ (as illustrated in Figure 1.1). Tensor $\mathcal{Y}$ is also expected to be in a compressed form for time and space efficiency. Other than the graphs, we are also given 1) a small amount of curated n-way associations, or 2) the curated bipartite associations between each pair of graphs when the n-way associations are unavailable. The learning task is formally defined as:

- ***Input graphs***:
  n different biological networks which are represented as undirected graphs $\{G^{(i)} = (V^{(i)}, E^{(i)}) : i = 1, \ldots, n\}$, where the number of nodes in graph $G^{(i)}$ is $|V^{(i)}| = I_i$, for $i = 1, \ldots, n$.

- ***Input associations:***

  - Option 1 (curated n-way associations): a small set $\{(i_1, i_2, \ldots, i_n) : i_j \in [1, I_j], \forall j = 1, \ldots, n\}$ of labeled n-way associations among the nodes across n graphs, where i_j denotes the node of graph $G^{(j)}$.
  - Option 2 (curated bipartite associations): a set $\{R_{i,j} \in \mathbb{R}_+^{I_i \times I_j} : \forall i, j \in [1, n] \text{ and } i < j\}$ of non-negative matrices, where $R_{i,j}$ contains the similarity

scores between the nodes of a pair of graphs $G^{(i)}$ and $G^{(j)}$.

- ***Learning task***:
  Given the input graphs and curated associations, predict the scores of all the n-way associations among the nodes across the n graphs in a compressed n-way tensor $\mathcal{Y} \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$.

1.3 Fundamental models

We will review a two categories of base models that have been previously proposed in literature to solve multi-relational learning problems. Our methods proposed in this thesis are developed based on these models.

- **Tensor decomposition**
  The first category of models [22, 23] rely on tensor decomposition technologies, where an incomplete sparse tensor of the observed multi-way associations is decomposed into low-dimensional components with respective to the prior knowledge carried in the networks. These components can then be used to estimate the missing part of the original tensor. Let $\mathcal{Y}^0 \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ be an n-way sparse tensor initialized with the labels of the observed n-way associations given in the set $\{(i_1, i_2, \ldots, i_n) : i_j \in [1, I_j], \forall j = 1, \ldots, n\}$ (described in Section 1.2 Option 1), with the missing associations represented as zeros. The goal is to learn a tensor $\mathcal{Y} \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ of complete n-way associations in a compressed representation. *Canonical Polyadic Decomposition (CPD)* and *Tucker Decomposition* are two widely adopted compressed representations of a tensor. As reviewed in [24] and Appendix A.1, CPD represents a tensor $\mathcal{Y} \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ as component matrices $\{A^{(i)} \in \mathbb{R}^{I_i \times r_i} : i = 1, \ldots, n\}$, where matrix $A^{(i)}$ is a low-rank representation of the i-th mode of the tensor $\mathcal{Y}$. In addition to the component matrices as in the CPD, there is also a low-dimensional core tensor $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times \ldots \times r_n}$ in Tucker decomposition, which encodes the level of interaction between the different components. In this thesis, we choose CPD as the compressed representation of the multi-relational tensors to maximize the scalability of our proposed methods for large-scale applications; we will also leverage the tensor computations in both CPD and Tucker to improve the computational efficiency of our methods.

- **Label propagation**

  The second category of methods [25–27] leverage the idea of label propagation on a product graph to predict association across the networks. Given a set of labeled multi-way tuples of graph nodes, these methods aim at labeling/scoring the unlabeled multi-way tuples by learning with the manifold structure in the product graph. In this thesis, we will focus on the label propagation model proposed in [28], which has been widely applied for graph-based semi-supervised learning and ranking problems. Given a graph $G = (V, E)$ with adjacency matrix W and N nodes, and a vector $\boldsymbol{y}^0 \in \mathbb{R}^{N\times 1}$ which contains the observed labels of (some) graph nodes, the label propagation learns the labeling scores of all the graph nodes in a vector $\boldsymbol{y} \in \mathbb{R}^{N\times 1}$ by performing the following iteration until convergence:

$$\boldsymbol{y}^{t+1} = \alpha S \boldsymbol{y}^t + (1-\alpha)\boldsymbol{y}^0, \tag{1.1}$$

  where t denotes the iteration number, and $S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ is the normalization of W by the degree matrix D. During each iteration, each node receives the information from its neighbors (first term), and also retains its initial information (second term). The hyperparameter $\alpha \in (0, 1)$ specifies the amount of information from the neighbors and the initial labels.

  The label propagation iteration in Equation (1.1) minimizes the following quadratic objective:

$$\begin{aligned}\mathcal{J}(\boldsymbol{y}) &= \frac{1}{2}\Big(\sum_{i,j=1}^{N} W_{ij}\Big(\frac{\boldsymbol{y}_i}{\sqrt{D_{ii}}} - \frac{\boldsymbol{y}_j}{\sqrt{D_{jj}}}\Big)^2 + \mu \sum_{i=1}^{N}(\boldsymbol{y}_i - \boldsymbol{y}_i^0)^2\Big) \\ &= \frac{1}{2}(\boldsymbol{y}^T(I - S)\boldsymbol{y} + \mu||\boldsymbol{y} - \boldsymbol{y}^0||^2).\end{aligned} \tag{1.2}$$

  The first term in Equation (1.2) is the *Laplacian regularization*, which ensures the adjacent nodes share similar labels. The second term is the *fitting constraint*, which preserves the initially observed labels. The trade-off between these two terms are determined by the hyperparameter $\mu = \frac{1-\alpha}{\alpha}$.

  It has been shown in [28], the iterations of label propagation in Equation (1.1) converges to the following closed-form solution which is the global optima of $\mathcal{J}(\boldsymbol{y})$ in Equation (1.2), based on the facts that eigenvalues of S are in $[-1, 1]$ and

$\alpha \in (0, 1)$.

$$\boldsymbol{y}^* = \lim_{t \to \infty} \boldsymbol{y}^t = (1 - \alpha)(I - \alpha S)^{-1} \boldsymbol{y}^0. \tag{1.3}$$

1.4 Challenges and objectives

There remain a number of challenges that hinder the application of the existing tensor-based methods to learn the multi-way associations among biological entities. Also, some emerging tasks in bioinformatics contain multi-way (tensor) structures in their data, have not been modeled using tensor technologies. We summarize the challenges and tasks as follows:

1. The existing tensor-based methods all require training with observed multi-way associations, which are however very scarce or even unavailable in many situations. The requirement severely limits the applicability of these methods to predicting high-order multi-way associations, in which bipartite associations among the biological entities widely exist in public available databases, while the curated multi-way associations are extremely sparse. Therefore, we expect a model to effectively utilize the observed bipartite associations, together with the networks to solve the multi-relational learning problem.

2. The label propagation methods are only empirically scalable to three-way tensors due to the necessity of computing the full tensor in every iteration. The exiting estimation approaches do not solve the scalability issue either, due to the suboptimal approximation of the product graph. On the other hand, the majority of tensor decomposition models are based on non-convex formulations, though applicable to multi-way tensors, can potentially lead to poor local minima especially for high-order tensors.

3. High-throughput spatial-transcriptomics RNA sequencing (sptRNA-seq) based on in-situ capturing technologies has recently been developed to spatially resolve transcriptome-wide mRNA expressions mapped to the captured locations in a tissue. The sptRNA-seq data, though has the 3D multi-way structure among spatial locations and genes, has not been modeled using tensor technologies to

complete the gene expression profiles and recover functionally relevant spatial patterns.

1.5 Contributions of the thesis

To solve the aforementioned challenges and tasks, we propose scalable machine learning methods to infer the multi-way associations among the network entities in compressed representations of the tensors, which are structured by the network manifolds. Our contributions are:

1. To effectively utilize the known bipartite associations without relying on the training multi-way associations, we propose an algorithm, named Graph-Regularized Tensor Completion from Observed Pairwise Relations (GT-COPR), to infer the multi-way associations across multiple biological networks in a low-rank tensor. GT-COPR regularizes tensor elements with the Laplacian of a single product graph and also requires the collapsed tensors to be consistent with the observed bipartite associations.

2. To scale up the label propagation algorithm to the product of multiple networks, we propose the Low-Rank Tensor-based Label Propagation (LowrankTLP) algorithm, which minimizes a novel optimization formulation by learning with a subset of eigen-pairs from the normalized product graph. Our formulation has the globally optimal solution, which minimizes an estimating error bound of recovering the true multi-relational tensor.

3. To leverage the multi-way associations between the genes and tissue locations in the recently developed sptRNA-seq data, we propose a graph-regularized tensor completion algorithm, named Fast Imputation of Spatially-Resolved Transcriptomes by Graph Regularized Tensor Completion (FIST) for imputing the missing mRNA expressions. Experimental results show that FIST not only significantly improves the imputation accuracy, but also captures the spatial characteristics in the gene expressions and reveals functions that are highly relevant to tissue types.

1.6 Outline of the thesis

The rest of this thesis will be organized into the following chapters:

- Chapter 2 first introduces the product graph regularization, then proposes graph-guided tensor completion models based on tensor decomposition and label propagation for the multi-relational learning task, and finally identifies their computational limitations which will be resolved in Chapters 3 and 4.
- Chapter 3 proposes the GT-COPR algorithm to infer the multi-way associations across multiple biological networks, by directly utilizing the bipartite associations, while not relying on the observed multi-way associations.
- Chapter 4 proposes the scalable LowrankTLP algorithm, which is based on a principled approximation to the closed-form solution of a classical label propagation model, enables propagating a high-order tensor on the product graph.
- Chapter 5 proposes the FIST algorithm based on the tensor decomposition model described in Chapter 2, to impute the missing values in the sptRNA-seq data by modeling the associations between genes and tissue locations with a 3-way tensor.
- Chapter 6 presents two error bounds to theoretically justify the optimization formulation of the LowrankTLP algorithm, and also analyzes the convergence of the CT-COPR and FIST algorithms.
- Chapter 7 summarizes the contributions and technologies of all the methods proposed in this thesis.

Chapter 2

Multi-relational Learning with Product Graphs and Tensor Completion

In Section 1.2, we proposed to formulate the multi-relational learning task as a tensor completion problem. The n-way associations across the nodes in n different graphs can be learned in an n-way tensor $\mathcal{Y}$ as illustrated in Figure 1.1. In this chapter, we propose novel graph-guided tensor completion methods, by regularizing the tensor $\mathcal{Y}$ using a single product graph. Our proposed methods are based on the tensor decomposition and label propagation models as previewed in Section 1.3. The rest of this chapter is organized as follows: Section 2.1 defines the notations of this thesis. Section 2.2 explains how the product graph integrates the heterogeneous information from the individual graphs, and can be utilized to regularize the n-way tensor. Section 2.3 proposes the tensor decomposition model with product graph regularization. Section 2.4 proposes the generalization of label propagation to the tensor product graph. Finally, in Section 2.5, we point out the challenges that limit the applicability of these methods to predicting high-order multi-way associations in real situations, which will be addressed in the forthcoming chapters.

2.1 Notations of the thesis

We summarize the notations in Table 2.1. Several useful definitions and lemmas which will be used in the derivations of this thesis are also given in Appendix A.1 and A.2. For Other general knowledge of tensor computations and technologies, we direct the readers to the survey paper [24].

Table 2.1: **Notations.**

Operand	Operator
Scalar: x	**Hadamard product:** $\circledast$
Vector: $\boldsymbol{x}$	**Khatri–Rao product:** $\odot$
Matrix: X	**Kronecker sum:** $\oplus$
Tensor : $\mathcal{X}$	**Kronecker product:** $\otimes$
Vectorization of a tensor: $\mathbf{vec}(\mathcal{X})$	**Vector outer product:** $\circ$
Mode-n matricization of a tensor: $X_{(n)}$	**n-mode product of a tensor:** $\times_n$

2.2 Regularization with product graph

In this section, we first define the construction of the product graphs from the individual graphs $\{G^{(i)} = (V^{(i)}, E^{(i)}) : i = 1, \ldots, n\}$, and then explain how the product graph topology integrates the connections in the original graphs for inferring the n-way tensor $\mathcal{Y}$. The following three types of product graphs [29] are considered in this thesis.

- ***Cartesian product graph (CPG)*** G^c: the pair of node tuples $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ in G^c are adjacent if and only if any pair of nodes a_i and b_i are adjacent in $G^{(i)}$ such that $(a_i, b_i) \in E^{(i)}$, and all the rest pairs are identical such that $a_j = b_j$ for all $j \neq i$. The adjacency and graph Laplacian matrices of G^c are obtained as $W^c = \oplus_{i=1}^n W^{(i)}$ and $L^c = \oplus_{i=1}^n L^{(i)}$ respectively.

- ***Tensor product graph (TPG)*** G^t: the pair of node tuples $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ in G^t are adjacent if and only if every pair of nodes a_i and b_i are adjacent in $G^{(i)}$ such that $(a_i, b_i) \in E^{(i)}$ for all $i = 1, \ldots, n$. The adjacency and graph Laplacian matrices of G^t are given by $W^t = \otimes_{i=1}^n W^{(i)}$ and $L^t = \otimes_{i=1}^n D^{(i)} - \otimes_{i=1}^n W^{(i)}$ respectively.

- ***Strong product graph (SPG)*** G^s: the pair of node tuples $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ in G^s are adjacent if and only if they are adjacent in either G^c or G^t. The adjacency and graph Laplacian matrices of G^s are obtained as $W^s = W^c + W^t$ and $L^s = L^c + L^t$ respectively.

Denoting $G = (V, E)$ as one of the three types of product graphs defined above, which integrates all the individual graphs $\{G^{(i)} = (V^{(i)}, E^{(i)}) : i = 1, \ldots, n\}$ as a single graph. The edges in graph G encode the similarities between the n-way tuples of graph nodes $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n), \forall\{a_l, b_l\} \in [1, I_l]$ as illustrated in Figure 1.1. Since the total number of nodes $|V| = \prod_{i=1}^{n} I_i$ in the product graph G is identical to the number of elements in an n-way tensor $\mathcal{Y} \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ with each individual graph matches with one mode in the tensor, the n-way relational learning problem can be converted to the graph labeling problem in product graph G. We aim to learn a tensor $\mathcal{Y}$ which contains the labels of the nodes in G. To solve the learning problem, we propose to regularize the whole tensor $\mathcal{Y}$ with the graph Laplacian L of product graph G by minimizing the quantity $\mathbf{vec}(\mathcal{Y})^T L \mathbf{vec}(\mathcal{Y})$, which is called *Laplacian regularization* or *smoothness constraint*. The learned n-way associations in $\mathcal{Y}$ are thus ensured to be smooth over the manifolds of product graph G, such that a pair of tensor elements $\mathcal{Y}_{a_n a_{n-1} \ldots a_1}$ and $\mathcal{Y}_{b_n b_{n-1} \ldots b_1}$ share similar values if the node tuples $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ in G are adjacent. Note that by definition, the strong and tensor product graphs have more edges compared with the Cartesian product graph, and thus might encode richer similarity information among the tensor elements, while the Cartesian product graph bridges two tensor elements under stricter condition which might incur less noise in defining the similarities.

2.3 Product graph regularized tensor decomposition

In the tensor decomposition model proposed below, a sparse tensor $\mathcal{Y}^0 \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ needs to be initialized with the labels of the observed n-way associations given in the set $\{(i_1, i_2, \ldots, i_n) : i_j \in [1, I_j], \forall j = 1, \ldots, n\}$ (described in Section 1.2 Option 1). Then the complete tensor $\mathcal{Y}$ of the inferred n-way associations can be obtained by solving the

optimization problem in Equation (2.1).

$$\begin{aligned} \underset{\{A^{(1)},A^{(2)},\ldots,A^{(n)}\}}{\text{minimize}} \quad & \frac{1}{2}||\mathcal{M} \circledast (\mathcal{Y}^0 - \mathcal{Y})||_{\mathcal{F}}^2 + \frac{\lambda}{2}\mathbf{vec}(\mathcal{Y})^T L \mathbf{vec}(\mathcal{Y}) \\ \text{subject to} \quad & \mathcal{Y} = [\![A^{(n)}, A^{(n-1)}, \ldots, A^{(1)}]\!], \end{aligned} \tag{2.1}$$

where $\lambda \in [0, 1]$ is a model hyperparameter, and $||.||_{\mathcal{F}}$ denotes the Frobenius norm of a tensor. The details about the model are explained below:

- **Consistency with the observations**

  $\mathcal{M}$ is a binary mask tensor to indicate the indices of the observed entries in $\mathcal{Y}^0$. The $(m_1, m_2, \ldots, m_n)$-th entry $\mathcal{M}_{m_1 m_2 \ldots m_n}$ which is defined below, represents whether the $(m_1, m_2, \ldots, m_n)$-th element in $\mathcal{Y}^0$ is observed or not.

$$\mathcal{M}_{m_1 m_2 \ldots m_n} = \begin{cases} 1 & \text{if } \mathcal{Y}^0_{m_1 m_2 \ldots m_n} \text{ is observed,} \\ 0 & \text{otherwise.} \end{cases}$$

  By introducing the squared-error $||\mathcal{M} \circledast (\mathcal{Y}^0 - \mathcal{Y})||_{\mathcal{F}}^2$ in the model, the inferred n-way tensor $\mathcal{Y}$ is ensured to be consistent with its observed counterparts in $\mathcal{Y}^0$.

- **Regularization with product graph**

  L in Equation (2.1) is the Laplacian matrix of the graph $G = (V, E)$, which can be any one of the three types of product graphs defined in Section 2.2. By introducing the *Laplacian regularization* term $\mathbf{vec}(\mathcal{Y})^T L \mathbf{vec}(\mathcal{Y})$ in equation (2.1), the inferred n-way associations in $\mathcal{Y}$ are ensured to be smooth over the manifolds of the product graph as has been explained in Section 2.2.

- **Compressed representation of the n-way tensor**

  It can be computationally expensive and often infeasible to compute or store a dense tensor $\mathcal{Y}$, especially in high-order applications. Therefore, we propose to compute an economy-size representation of $\mathcal{Y}$ via introducing the equality constraint $\mathcal{Y} = [\![A^{(n)}, A^{(n-1)}, \ldots, A^{(1)}]\!]$, which is called *canonical polyadic decomposition (CPD)* [24] of $\mathcal{Y}$ defined below

$$\mathcal{Y} = [\![A^{(n)}, A^{(n-1)}, \ldots, A^{(1)}]\!] = \sum_{i=1}^{r} [A^{(n)}]_{:,i} \circ [A^{(n-1)}]_{:,i} \circ \cdots \circ [A^{(1)}]_{:,i}, \tag{2.2}$$

where r is the rank of $\mathcal{Y}$, and $[A^{(j)}]_{:,i}$ denotes the i-th column of the low-rank matrix $A^{(j)} \in \mathbb{R}^{I_j \times r}$, for all $j = 1, \ldots, n$. By utilizing the tensor CPD-form, we replace the optimization variable tensor $\mathcal{Y}$ with matrices $\{A^{(1)}, A^{(2)}, \ldots, A^{(n)}\}$, reducing the number of parameters from $\prod_{i=1}^{n} I_i$ to $r(\sum_{i=1}^{n} I_i)$.

2.4 Label propagation on tensor product graph

In this section, we generalize the graph-based semi-supervised learning algorithm, named label propagation, proposed in [28] to tensor product graph (TPG) for the multi-relational learning task defined in Section 1.2. We first generalize the optimization framework of label propagation to the normalized TPG to model the learning task. Next, we propose the tensor-based label propagation algorithm to solve the optimization problem and gives the closed-form solution.

2.4.1 Regularization framework with normalized TPG

We propose to solve the multi-relational learning problem, i.e., the graph labeling problem described in Section 2.2 by generalizing the regularization framework of the label propagation model in [28] given in Equation (1.2) to tensor product graph (TPG) as the following,

$$\mathcal{J}(\mathcal{Y}) = \frac{1}{2}(vec(\mathcal{Y})^T(I - S)vec(\mathcal{Y}) + \mu||vec(\mathcal{Y}) - vec(\mathcal{Y}^0)||_2^2). \tag{2.3}$$

The first term of $\mathcal{J}(\mathcal{Y})$ in Equation (2.3) is is analog to *Laplacian regularization* term defined in Section 2.2 and Equation (2.1), where $I - S$ is the *normalized graph Laplacian* of the TPG $G = (V, E)$ with the adjacency matrix W, to ensure a pair of tensor elements (inferred multi-way associations) $\mathcal{Y}_{a_n a_{n-1} \ldots a_1}$ and $\mathcal{Y}_{b_n b_{n-1} \ldots b_1}$ share similar scores if the degree-normalized edge weight in graph G is large. Here, $S = [D]^{-\frac{1}{2}} W [D]^{-\frac{1}{2}}$ is the normalized TPG which is derived as

$$\begin{aligned} S &= (\otimes_{i=1}^{n} [D^{(i)}]^{-\frac{1}{2}})(\otimes_{i=1}^{n} W^{(i)})(\otimes_{i=1}^{n} [D^{(i)}]^{-\frac{1}{2}}) && (2.4) \\ &= \otimes_{i=1}^{n} ([D^{(i)}]^{-\frac{1}{2}} W^{(i)} [D^{(i)}]^{-\frac{1}{2}}) && (2.5) \\ &= \otimes_{i=1}^{n} S^{(i)}, \end{aligned}$$

where Equation (2.4) is obtained by the fact that the degree matrices are diagonal. Equation (2.5) is obtained by Lemma A.2.1 in appendix. The second term of $\mathcal{J}(\mathcal{Y})$ in Equation (2.3) is called the *fitting constraint*, which penalizes the difference between the inferred tensor $\mathcal{Y}$ and the initial observation $\mathcal{Y}^0$. $\mu > 0$ is a hyperparameter weighting the two terms.

2.4.2 Tensor-based label propagation algorithm

The objective function $\mathcal{J}(\mathcal{Y})$ in Equation (2.3) can be minimized by performing the fixed-point iteration in Equation (2.6), which is a generalization of the label propagation iteration given by Equation (1.1) to TPG as follows,

$$vec(\mathcal{Y}^{t+1}) = \alpha(\otimes_{i=1}^{n} S^{(i)})vec(\mathcal{Y}^t) + (1-\alpha)vec(\mathcal{Y}^0), \tag{2.6}$$

where $\alpha = \frac{1}{1+\mu} \in (0,1)$ is a balancing hyperparameter and t denotes the iteration number. According to the *vectorization property* of Tucker decomposition (Definition A.1.2 in appendix), Equation (2.6) can be rewritten as the following matrix-tensor form

$$\mathcal{Y}^{t+1} = \alpha\mathcal{Y}^t \times_1 S^{(n)} \times_2 S^{(n-1)} \cdots \times_n S^{(1)} + (1-\alpha)\mathcal{Y}^0. \tag{2.7}$$

Since the eigenvalues of S are in $[-1,1]$ and $\alpha \in (0,1)$, following Equation (1.3), the iteration in Equation (2.6) can be shown to converge to the closed-form solution of $\mathcal{J}(\mathcal{Y})$ as

$$vec(\mathcal{Y}^*) = \lim_{t\to\infty} vec(\mathcal{Y}^t) = (1-\alpha)(I-\alpha S)^{-1}vec(\mathcal{Y}^0). \tag{2.8}$$

Furthermore, given the eigen-decomposition of each $S^{(i)}$ as $\{S^{(i)} = Q^{(i)}\Lambda^{(i)}Q^{(i)T} : i = 1,\ldots,n\}$, the eigen-decomposition of S can be expressed as

$$S = Q\Lambda Q^T = (\otimes_{i=1}^{n} Q^{(i)})(\otimes_{i=1}^{n}\Lambda^{(i)})(\otimes_{i=1}^{n} Q^{(i)T}),$$

according to Lemmas A.2.1 and A.2.3 in appendix. Substituting S into Equation (2.8) we have

$$vec(\mathcal{Y}^*) = (1-\alpha)(\otimes_{i=1}^{n} Q^{(i)})(I-\alpha(\otimes_{i=1}^{n}\Lambda^{(i)}))^{-1}(\otimes_{i=1}^{n} Q^{(i)T})vec(\mathcal{Y}^0). \tag{2.9}$$

2.5 Computational challenges

In real scenarios, there are two main challenges that severely limit the application scopes of either the tensor decomposition or label propagation for high-order multi-relational learning, as summarized below.

1. Both the tensor decomposition and label propagation models require training with a sparse tensor $\mathcal{Y}^0$, initialized by the set $\{(i_1, i_2, \ldots, i_n) : i_j \in [1, I_j], \forall j = 1, \ldots, n\}$ (Option 1) of labeled n-way associations. Unfortunately, the n-way associations are unavailable in many applications. In contrast, the curated bipartite associations $\{R_{i,j} \in \mathbb{R}_+^{I_i \times I_j} : \forall i, j \in [1, n] \text{ and } i < j\}$ (Option 2) widely exist in public databases. Therefore, we expect novel variations of both models to enable utilizing the bipartite associations, without replying on the inital n-way associations.

2. The tensor formulation of label propagation is computationally intensive to solve. Even if the initialization $\mathcal{Y}^0$ is in a sparse form, the density of tensor $\mathcal{Y}$ increases exponentially in each iteration of the label propagation in Equation (2.7). Therefore, the space complexity is $O(\prod_{i=1}^{n} I_i)$ for store the dense tensor and the time complexity is $O((\prod_{i=1}^{n} I_i)(\sum_{i=1}^{n} I_i))$ per iteration. On the other hand, computing the closed-form solution in Equation (2.9) from right to left needs $2n$ matrix-tensor products in total with the *vectorization property* of Tucker decomposition (Definition A.1.2 in appendix). Therefore, the space and time complexity will be the same as running two iterations of label propagation, apart from computing the eigen-decompositions of all the normalized graphs $\{S^{(i)} : i = 1, \ldots, n\}$.

In Chapter 3 and 4, we propose two ways to solve the first challenge. In Chapter 3, we propose a general tensor-based optimization framework and the Graph-Regularized Tensor Completion from Observed Pairwise Relations (GT-COPR) algorithm to directly infer the multi-way associations from the observed bipartite associations. Then in Chapter 4, we propose to convert the bipartite associations to an initial tensor $\mathcal{Y}^0$, in the rank-r CPD-form as in Equation (2.2). To tackle the second challenge, we propose the Low-Rank Tensor-based Label Propagation algorithm (LowrankTLP), based on a theoretically justified approximation of the linear transformation matrix $(I - \alpha S)^{-1}$ in the closed-form solution in Equation (2.8), enables computing label propagation of a high-order n-way tensor on TPG.

Chapter 3

Multi-relational Learning from Bipartite Associations

3.1 Introduction

Inferring the *disease-gene-chemical* multi-way associations based on the network topologies can guide the development and application of therapies for precision medicine [9]. A real example of known pharmacogenomic multi-way associations across three biomedical subnetworks is shown in Figure 3.1, e.g. the multi-way association "acute myeloid leukemia"-"NRAS"-"fedratinib" indicates mutation of NRAS (gene) can impact the sensitivity of fedratinib (chemical) in treating acute myeloid leukemia (disease) [9].

Current studies mainly emphasize on imputing unknown bipartite associations among biomedical and biological networks, based on the observed associations and network topological information. For example, [31] and [32] predict novel *drug-target* associations for drug repositioning using observed associations obtained from publicly available databases, together with the drug structural and target sequence similarity networks; [17] and [33] adopt network propagation based approaches according to the smoothness assumptions made on the protein-protein interaction (PPI) network and phenotype similarity networks to identify *disease-gene* pairs for prioritizing disease causal genes. To infer the *disease-gene*, *disease-chemical* and *gene-chemical* bipartite associations simultaneously, [10] formulates a collective matrix completion problem with

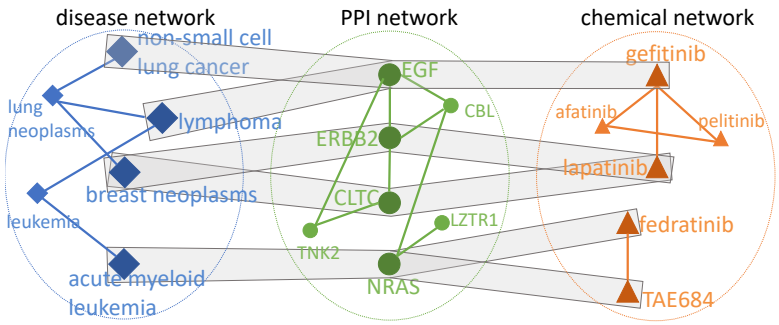


Figure 3.1: **Pharmacogenomic multi-way associations.**
The gray boxes contain the significant *disease-gene-chemical* multi-way associations ($p < 0.001$, FDR < 0.25) identified in [9] through ANOVA analysis. The solid lines denote the phenotype-based *disease-disease* pairs found in [4], *gene-gene* pairs reported by BioGRID v3.5 [30] and *chemical-chemical* pairs with the Tanimoto coefficients between the 881-bit structure fingerprints above 0.75.

graph regularization, and [34] applies network diffusion algorithms on the six-modal network constructed by stacking all the bipartite relational matrices and networks from the three data domains.

To learn the *disease-gene-chemical* multi-way associations directly, we propose a general product graph regularized tensor completion framework. Our goal is to predict the unknown n-way associations based on the topological information carried in variant types of product graphs. In the last decade, graph-based tensor completion techniques have received great interest. In [25–27], semi-supervised manifold learning technology referred as label propagation [35–37] is applied for completing the partially observed tensors. Specifically, [25] predicts the link types of the unknown hyperlinks among knowledge graphs in a tensor based on the conjugate gradient descent optimization, whose scalability is later improved in [26] through low-rank approximation of the knowledge graphs; [27] introduces the product graph regularization into the tensor completion objective via a Gaussian random fields prior to infer the cross-graph multi-way associations. As the graph-regularized matrix factorization [38–40] has been extensively explored and achieved substantial success in data mining areas from collaborative filtering to link prediction, [41] generalized the idea to tensor completion via decomposing

an incomplete tensor into low-rank matrices with their values jointly smoothing over the manifolds of the tensor product graph. When dealing with the temporal-bipartite-relational tensor completion problem, [42] proposes to collapse the subtensor of previous time stamps into a matrix (bipartite graph), followed by applying the well known Katz measure for bipartite link prediction at a future time stamp. Similar tensor collapsing idea has also been adopted in [43] for nodes classification on the temporal bipartite graphs.

As reviewed above, the existing tensor completion methods all require training with observed multi-way associations, which are however very scarce or even unavailable in many situations, especially in high-order multi-way associations. The requirement severely limits the applicability of these methods to predicting *disease-gene-chemical* associations, in which bipartite associations among *disease-gene*, *disease-chemical* and *gene-chemical* widely exist in public available databases such as CTD [44], DrugBank [45] and ChEMBL [46], while the curated triple-wise associations are extremely sparse. Therefore, we consider utilizing the observed bipartite associations, together with the knowledge graphs to solve the multi-relational learning problem. In this chapter, we formally establish a novel and general tensor-based optimization objective and a scalable iterative method GT-COPR (Graph-Regularized Tensor Completion from Observed Pairwise Relations) to efficiently infer the n-way associations in a compressed tensor. The key novelties of our model are:

- We propose to apply tensor collapsing to capture the cross-mode dependencies and the global consistencies with the observed bipartite associations exist in database.
- We propose to co-regularize tensor elements with the Laplacian of three types of product graphs (Cartesian, tensor and strong product) to introduce the local consistencies among the n-way associations in multiple entities.
- We propose to learn a compressed tensor in CPD-form to guarantee the space and time efficiencies for learning high-order multi-way associations.

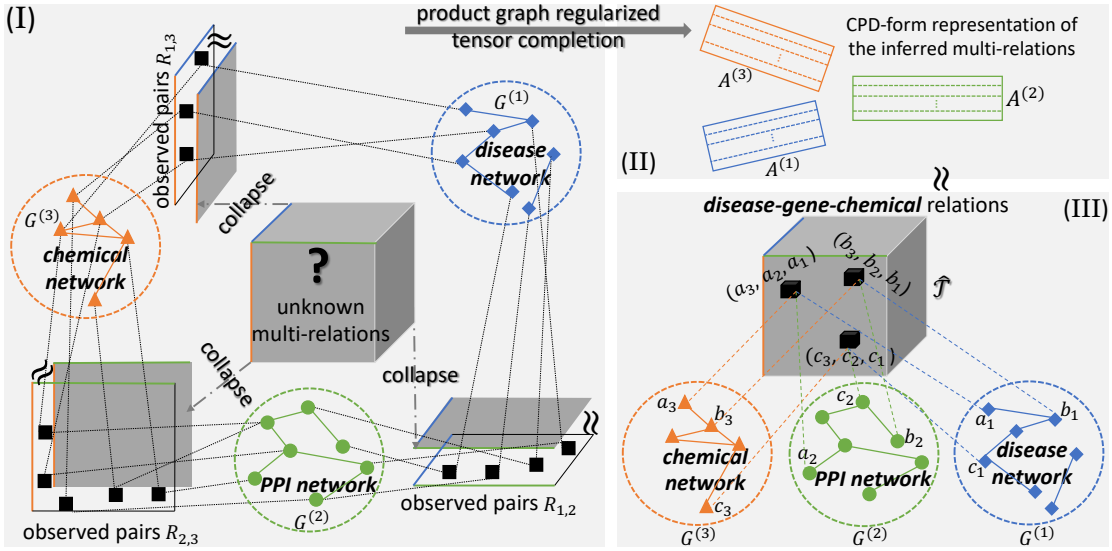


Figure 3.2: **Overview of GT-COPR algorithm (explained by 3-way tensor).** Based on the topological information carried in the Cartesian, tensor or strong product of the knowledge graphs $G^{(1)}, G^{(2)}$ and $G^{(3)}$, and the consistencies between the collapsed tensors and the observed bipartite associations $R_{1,2}, R_{1,3}$ and $R_{2,3}$ among the nodes of every pair of graphs [part (I)], the proposed GT-COPR algorithm can learn a low-rank CPD-form representation [part (II)] of the three-way relational tensor $\mathcal{T}$ [part (III)] which predicts the cross-graph multi-way associations.

3.2 Tensor-based multi-relational learning

As illustrated in Figure 3.2, our learning task is to infer the multi-way associations across the nodes of n knowledge graphs $\{G^{(i)} = (V^{(i)}, E^{(i)}) : |V^{(i)}| = I_i, \forall i = 1, \ldots, n\}$ in an n-way tensor $\mathcal{T} \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ (Figure 3.2 (III)), given a set of non-negative matrices $\{R_{i,j} \in \mathbb{R}_+^{I_i \times I_j} : \forall i, j \in [1, n] \text{ and } i < j\}$ (Figure 3.2 (I)) with $R_{i,j}$ holding the observed bipartite associations between the nodes of graphs $G^{(i)}$ and $G^{(j)}$ with zeros representing the unknown associations. To solve the multi-relational learning problem, we first propose our optimization formulation, and then present an efficient iterative algorithm GT-COPR to minimize the optimization objective. The convergence of GT-COPR will be proven in Section 6.2.

3.2.1 Optimization formulation

Our key ideas of solving the n-way relational learning problem are 1) n-way associations inferred in the tensor $\mathcal{T}$ are required to be consistent with each other by the connectivity in the product graph defined in Section 2.2; 2) the collapsed tensors are required to be consistent with the corresponding bipartite relational matrices; and 3) the inferred tensor $\mathcal{T}$ is compressed in its CPD-form for space and time efficiency, together in a novel optimization formulation presented below in Proposition 3.2.1.

Proposition 3.2.1. *The tensor $\mathcal{T} \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ of inferred n-way associations can be approximated and compressed in the rank-r CPD-form $\hat{\mathcal{T}} = [\![A^{(n)}, A^{(n-1)}, \ldots, A^{(1)}]\!]$, which is obtained by solving the following optimization problem:*

$$\begin{aligned} \underset{\{A^{(i)}: i=1,\ldots,n\}}{\textit{minimize}}\ \mathcal{J} = & \sum_{i,j:\ i<j} \frac{1}{2} ||R_{i,j} - \frac{1}{\prod_{l \neq i,j} I_l} \mathbf{collapse}(\hat{\mathcal{T}}, i, j)||_F^2 \\ & + \frac{\lambda}{2} \mathbf{vec}(\hat{\mathcal{T}})^T L \mathbf{vec}(\hat{\mathcal{T}}) + \frac{\beta}{2} \sum_{i=1}^{n} ||A^{(i)}||_F^2 \\ \textit{subject to} \quad & A^{(i)} \geq 0, \forall i = 1, \ldots, n, \end{aligned} \tag{3.1}$$

where $\mathbf{collapse}(\hat{\mathcal{T}}, i, j)$ *denotes collapsing tensor $\hat{\mathcal{T}}$ into an $I_i \times I_j$ matrix by summing over the tensor slices along the corresponding modes (illustrated in Figure 3.2 (I)); L is the Laplacian matrix of the graph $G = (V, E)$, which can be any one of the three types of product graphs defined in Section 2.2; λ and $\beta \in (0, 1)$ are hyperparameters.*

Equation (3.1) is a variation of the tensor decomposition objective given in Equation (2.1), enables using the observed bipartite associations as input (Option 2 in Section 1.2). The first term of the objective function $\mathcal{J}$ in Equation (3.1) requires averaging over the slices of tensor $\hat{\mathcal{T}}$ to be globally consistent with the observed bipartite relational matrices. The second term is the regularization with product graph as discussed in Section 2.2. We consider all the three types of product graph manifolds in our model since both Cartesian and tensor product similarities exist in the real biomedical and biological networks as shown in Figure 3.1, and Cartesian product graph also has a strong capability of jointly clustering the topologically related objects from multiple data domains as will be shown in the experiments later. The third term is the standard *Tikhonov regularization* which penalizes overly complex model to avoid over-fitting; and

the CPD-form of $\hat{\mathcal{T}}$ guarantees the inferred n-way associations to be in a compressed form with low space complexity $\mathrm{O}(r\sum_i I_i)$.

Algorithm 3.1: GT-COPR (for strong product graph)

Data: 1) Knowledge graphs $\{G^{(i)} : i = 1, \dots n\}$, 2) observed bipartite relational matrices $\{R_{i,j} \in \mathbb{R}_+^{I_i \times I_j} : \forall i, j \in [1, n] \text{ and } i < j\}$, 3) hyper parameters λ and β, and 4) randomly initialized non-negative low-rank matrices $\{A^{(i)} : i = 1, \dots, n\}$ with rank r.

Result: A CPD-form tensor $\hat{\mathcal{T}} = [\![A^{(n)}, A^{(n-1)}, \dots A^{(1)}]\!]$ which stores the inferred n-way associations.

```
while not converge do
    for i = 1 to n do
        Update A^(i) by the rule given in Theorem 3.2.1.
    end
end
Return {A^(i) : i = 1, ..., n}
```

3.2.2 GT-COPR algorithm

The objective function $\mathcal{J}$ in Equation (3.1) is non-convex on variables $\{A^{(i)} : i = 1, \dots, n\}$ jointly, thus finding its global minimum is difficult. In the following, we propose an efficient iterative algorithm Graph-Regularized Tensor Completion from Observed Pairwise Relations (GT-COPR) as summarized in Algorithm 3.1 to find the local minimum of $\mathcal{J}$ based on the multiplicative updating rule structurally similar to the method for solving non-negative matrix factorization (NMF) problems [38,47]. Without loss of generality, we only show the derivations for the strong product graph regularization. As introduced in Section 2.2, the Laplacian matrix of the strong product graph is given by

$$L = L^c + L^t = \oplus_{i=1}^n L^{(i)} + \otimes_{i=1}^n D^{(i)} - \otimes_{i=1}^n W^{(i)}.$$

The derivations can be easily degenerated for the Cartesian (L^c) or tensor product (L^t) graph regularized GT-COPR with small modifications.

For simplicity, we rewrite the objective function in Equation (3.1) as $\mathcal{J} = \mathcal{J}_1 +$

Table 3.1: **Auxiliary variables.**

Variable	Form of variable	Time complexity
$\Phi_j^{(-i)}$	$((A^{(j)T}A^{(j)}) \underset{l\neq i,j}{\circledast} (\boldsymbol{a}_l\boldsymbol{a}_l^T))/(\prod_{k\neq i,j} I_k^2)$	$O(K^2 I_j)$
$\Phi_{j,k}^{(-i)}$	$((A^{(j)T}A^{(j)}) \circledast (A^{(k)T}A^{(k)}) \underset{l\neq i,j,k}{\circledast} (\boldsymbol{a}_l\boldsymbol{a}_l^T))/(\prod_{m\neq j,k} I_m^2)$	$O(K^2(I_j + I_k))$
$\Theta_j^{(-i)}$	$R_{i,j}(\underset{l\neq i,j}{\odot} \boldsymbol{a}_l^T \odot A^{(j)})/(\prod_{k\neq i,j} I_k)$	$O(K\lvert R_{i,j}\rvert)$
$\Theta_{j,k}^{(-i)}$	$\mathbf{1}_i\mathbf{vec}(R_{j,k})^T(A^{(k)} \odot A^{(j)} \underset{l\neq i,j,k}{\odot} \boldsymbol{a}_l^T)/(\prod_{m\neq j,k} I_m)$	$O(K\lvert R_{j,k}\rvert)$
$\Psi_j^{(-i)}$	$(A^{(j)T}L^{(j)}A^{(j)}) \underset{l\neq i,j}{\circledast} (A^{(l)T}A^{(l)})$	$O(K\lvert W^{(j)}\rvert + K^2(\sum_{l\neq i} I_l))$

$\lambda\mathcal{J}_2 + \beta\mathcal{J}_3$. Let $\boldsymbol{a}_i^T = \mathbf{1}_i^T A^{(i)}$ be the row summation of matrix $A^{(i)}$, we first define five auxiliary variables in Table 3.1 which are required for the derivations.

By expanding the tensor collapsing operator in $\mathcal{J}_1$ as:

$$\begin{aligned}&\mathbf{collapse}(\llbracket A^{(n)}, A^{(n-1)}, \ldots, A^{(1)}\rrbracket, i, j) = \\ &\llbracket \boldsymbol{a}_1^T, \ldots, \boldsymbol{a}_{i-1}^T, A^{(i)}, \boldsymbol{a}_{(i+1)}^T, \ldots, \boldsymbol{a}_{(j-1)}^T, A^{(j)}, \boldsymbol{a}_{(j+1)}^T, \ldots, \boldsymbol{a}_n^T\rrbracket,\end{aligned}$$

we obtain the partial derivative of the first objective term in a linear form of $A^{(i)}$ as:

$$\frac{\partial \mathcal{J}_1}{\partial A^{(i)}} = -(\sum_{j\neq i} \Theta_j^{(-i)} + \sum_{j,k\neq i:\ j<k} \Theta_{j,k}^{(-i)}) + \mathbf{1}_i\mathbf{1}_i^T A^{(i)}(\sum_{j,k\neq i:\ j<k} \Phi_{j,k}^{(-i)}) + A^{(i)}(\sum_{j\neq i} \Phi_j^{(-i)}).$$

Next, we expand the second objective term as follows

$$\begin{aligned}\mathcal{J}_2 &= \frac{1}{2}\mathbf{vec}(\llbracket A^{(n)}, \ldots, A^{(1)}\rrbracket)^T L\mathbf{vec}(\llbracket A^{(n)}, \ldots, A^{(1)}\rrbracket) \\ &= \frac{1}{2}\mathbf{1}^T(\odot_{i=1}^n A^{(i)})^T L(\odot_{i=1}^n A^{(i)})\mathbf{1} \qquad (3.2) \\ &= \frac{1}{2}\mathbf{1}^T(\underset{i}{\circledast}(A^{(i)T}D^{(i)}A^{(i)}) - \underset{i}{\circledast}(A^{(i)T}W^{(i)}A^{(i)}) \\ &\quad + \sum_{i=1}^n (A^{(i)T}L^{(i)}A^{(i)}) \underset{l\neq i}{\circledast} (A^{(l)T}A^{(l)}))\mathbf{1}. \qquad (3.3)\end{aligned}$$

Equation (3.2) holds by the *Vectorization property* of the CPD-form given in Definition A.1.1 in appendix. Lemmas A.2.1 and A.2.2 in appendix are applied to obtain Equation (3.3). The partial derivative of $\mathcal{J}_2$ to $A^{(i)}$ is then obtained as

$$\begin{aligned}&\frac{\partial \mathcal{J}_2}{\partial A^{(i)}} = L^{(i)}A^{(i)}(\underset{j\neq i}{\circledast}(A^{(j)T}A^{(j)})) + A^{(i)}(\sum_{j\neq i} \Psi_j^{(-i)}) + \\ &D^{(i)}A^{(i)}(\underset{j\neq i}{\circledast}(A^{(j)T}D^{(j)}A^{(j)})) - W^{(i)}A^{(i)}(\underset{j\neq i}{\circledast}(A^{(j)T}W^{(j)}A^{(j)})).\end{aligned}$$

Combining $\frac{\partial \mathcal{J}_1}{\partial A^{(i)}}$, $\frac{\partial \mathcal{J}_2}{\partial A^{(i)}}$ and $\frac{\partial \mathcal{J}_3}{\partial A^{(i)}} = A^{(i)}$, we obtain the partial derivative of $\mathcal{J}$ to component matrix $A^{(i)}$ as the following linear form:

$$\frac{\partial \mathcal{J}}{\partial A^{(i)}} = -X^1 - X^2 A^{(i)} X^3 + X^4 A^{(i)} X^5 + X^6 A^{(i)} X^7 + A^{(i)} X^8 + \beta A^{(i)}, \tag{3.4}$$

where the matrices $X^1 = \sum_{j \neq i} \Theta_j^{(-i)} + \sum_{j,k \neq i:\ j<k} \Theta_{j,k}^{(-i)}$, $X^2 = \lambda W^{(i)}$, $X^3 = \circledast_{j \neq i} (A^{(j)T} A^{(j)}) + \circledast_{j \neq i} (A^{(j)T} W^{(j)} A^{(j)})$, $X^4 = \lambda D^{(i)}$, $X^5 = \circledast_{j \neq i} (A^{(j)T} A^{(j)}) + \circledast_{j \neq i} (A^{(j)T} D^{(j)} A^{(j)})$, $X^6 = \mathbf{1}_i \mathbf{1}_i^T$, $X^7 = \sum_{j,k \neq i:\ j<k} \Phi_{j,k}^{(-i)}$, $X^8 = \sum_{j \neq i} (\Phi_j^{(-i)} + \lambda \Psi_j^{(-i)})$ are all non-negative. According to Equation (3.4), we present Theorem 3.2.1 below to provide the updating rule of the proposed GT-COPR algorithm. The convergence of GT-COPR has theoretical guarantee as will be discussed in Section 6.2.

Theorem 3.2.1. *Updating variables $\{A^{(i)} : i = 1, \ldots, n\}$ alternatively according to the rule given below can monotonically decrease the objective function $\mathcal{J}$ until they converge to the fixed-point solution.*

$$A_{ab}^{(i)} \leftarrow \frac{A_{ab}^{(i)} (X^1 + X^2 A^{(i)} X^3)_{ab}}{(X^4 A^{(i)} X^5 + X^6 A^{(i)} X^7 + A^{(i)} X^8 + \beta A^{(i)})_{ab}}$$

3.2.3 Time and space complexities

Assuming that $n < I_i$ and $r < I_i$, for all $i = 1, \ldots, n$, and with a slight abuse of notation by denoting $|S|$ also as the number of non-zeros in matrix S, we summarize the time complexities of computing the five auxiliary variables in Table 3.1. Based on it, we obtain the time complexity required to compute X^1 as $O(\sum_{j,k:\ j<k} K|R_{j,k}|)$, to compute $X^2 A^{(i)} X^3$ and $A^{(i)} X^8$ as $O(\sum_j (K^2 I_j + K|W^{(j)}|))$, and to compute $X^4 A^{(i)} X^5$ and $X^6 A^{(i)} X^7$ as $O(\sum_j (K^2 I_j))$. Thus, the overall time complexity of updating the i-th component matrix $A^{(i)}$ in one iteration of GT-COPR is $O(\sum_{j,k:\ j<k} K|R_{j,k}| + \sum_j (K^2 I_j + K|W^{(j)}|))$. The space required to store the bipartite relational matrices is $O(\sum_{j,k:\ j<k} |R_{j,k}|)$, to store the networks is $O(\sum_j |W^{(j)}|)$, and to store the component matrices is $O(\sum_j K I_j)$. Thus, the overall space complexity is $O(\sum_{j,k:\ j<k} |R_{j,k}| + \sum_j (|W^{(j)}| + K I_j))$. Though we only show the derivation for the strong product graph regularization, it is not hard to observe

that the three types of product graphs share the same theoretical time and space complexities. Note that the regularization with the tensor product graph does not involve the $\Psi_j^{(-i)}$ term in Table 3.1, and thus can be empirically faster than using Cartesian or strong product regularization.

3.3 Experiments

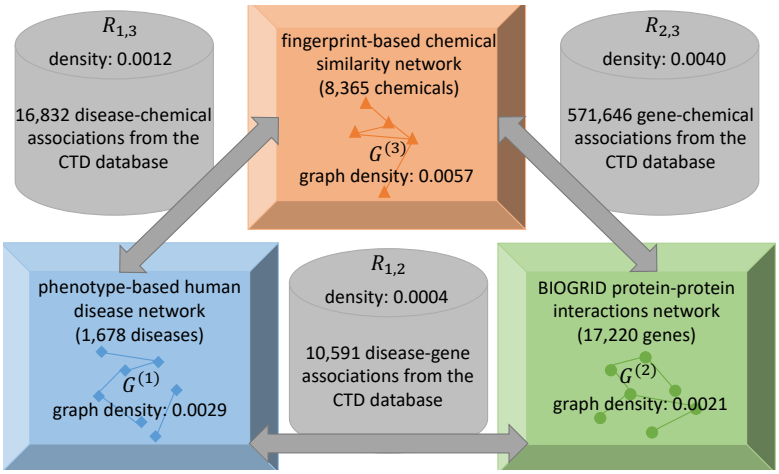


Figure 3.3: **Experimental data integrated from multiple sources.**

In this section we first describe the datasets that we integrate for learning *disease-gene-chemical* associations. Then we compare the prediction performance of GT-COPR with other methods through tensor fiber-wise and slice-wise evaluations. To show the potential clinical value of GT-COPR, we further validate the inferred triple-wise associations with the significant cancer-specific pharmacogenomic interactions reported by [9] from the analysis of the Genomics of Drug Sensitivity in Cancer (GDSC) [48] cell line dataset. Finally, we perform statistical analysis on the learned component matrices to show that GT-COPR also detects topologically and biologically relevant disease, gene and chemical components.

3.3.1 Data integration

We integrate multiple data sources to build the bipartite associations and knowledge graphs to infer the *disease-gene-chemical* associations. We first downloaded the bipartite associations from the Comparative Toxicogenomics Database (CTD) [44], which provides manually curated associations between chemicals, diseases and genes extracted from the published literature, with chemicals and diseases represented as Medical Subject Headings (MeSH) terms and genes represented as official gene symbols. Next, to obtain the networks we 1) downloaded the *Homo sapiens* protein-protein interactions (PPI) network from BioGRID 3.5 [30] as the *gene-gene* network; 2) we downloaded the human common disease network [4] as our *disease-disease* network where two common diseases are connected if their Human Phenotype Ontology (HPO) based phenotypic profiles share a high similarity; and 3) we construct the *chemical-chemical* network by first converting those CTD chemicals to PubChem 881-bit structure fingerprints using ChemRICH database [7], then add an edge between every pair of chemicals if the Tanimoto coefficient between their fingerprints is above 0.75 as suggested by [7]. The statistics of the integrated dataset are summarized in Figure 3.3, which includes the data sources, numbers of graph nodes, numbers of bipartite associations, and densities of the networks and bipartite relational matrices.

3.3.2 Methods for comparison

As mentioned in Section 3.1, there is no tensor-based method developed for inferring cross-graph multi-way associations from the observed bipartite associations. To benchmark the performance of GT-COPR, we compared it with five graph-based non-negative matrix factorization (NMF) methods. 1) wiZAN-Dual [40]: a dual graph regularized NMF with weight and imputation matrices in the objective, which was applied for prediction of *drug-target* interactions in [32]. 2) GWNMF [39]: a dual graph regularized NMF with the binary weight matrix indicating the observed and unobserved bipartite associations. 3) GWNMTF [39]: an alteration of the GWNMF to non-negative matrix tri-factorization. 4) FASCINATE [10]: a generalization of wiZAN-Dual to the joint factorization of multiple matrices, which was applied to infer the *disease-gene*, *disease-chemical* and *gene-chemical* bipartite associations simultaneously. 5) SNMF: symmetric

NMF [49] applied on the matrix constructed by putting all graphs on the diagonal blocks and all bipartite relational matrices on the off-diagonal blocks.

We choose α from $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ and fix $\beta = 0.1$ for GT-COPR. The graph hyperparameters of wiZAN-Dual/FASCINATE and GWNMF/GWNMTF are set by searching the grids {0.1, 0.5, 0.9, 1} and {0.1, 1, 10, 100} respectively, as suggested in their papers. Note that GT-COPR and the baseline methods use different scales of graph hyperparameters since the gradients of their variables are in different scales. To determine the rank r of the component matrices, we plot the top-1000 sorted singular values of each bipartite relational matrices in Figure 3.4. Interestingly, we can observe that the spectral energy of each of the three bipartite relational matrices is dominated by their top-200 singular values. Therefore, we set r to be the "elbow point" 200 for all the methods.

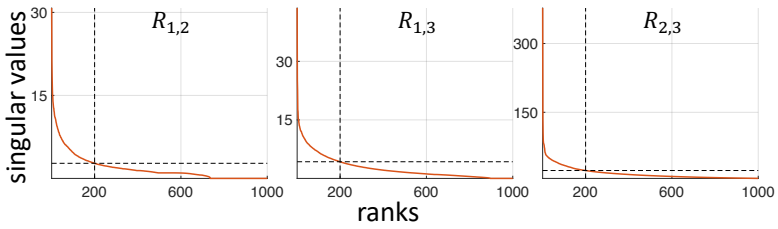


Figure 3.4: **Elbow plots of the singular values of the bipartite matrices.**

3.3.3 Evaluation of the predictive performance

To evaluate the performance of predicting the *disease-gene-chemical* triples, we first construct a 3-way binary ground truth tensor $\mathcal{T}$ with $\mathcal{T}_{ijk}$ denoting the association between the i-th disease, j-th gene and k-th chemical, which is positive if at least two interactions among the triple are observed in the bipartite relational matrices. Then we evaluate the performances for predicting the *disease-gene-chemical* triples by disease, gene and chemical tensor fibers and slices respectively. Without loss of generality, we use the disease fiber and slice as examples to explain our evaluation procedures as follows.

- ***Fiber-wise evaluation***: When evaluating the disease fiber $\mathcal{T}_{:jk}$ we first eliminate

all the connections to the j-th gene and k-th chemical from the *disease-gene* and *disease-chemical* relational matrices respectively; next, we construct $\hat{\mathcal{T}}_{:jk}$ using the learned low-rank matrices as described in Proposition 3.2.1, which is then treated as a score vector to predict $\mathcal{T}_{:jk}$.

- ***Slice-wise evaluation***: When evaluating the disease slice $\mathcal{T}_{i::}$ we first eliminate all the connections to the i-th disease from the *disease-gene* and *disease-chemical* relational matrices respectively; next, we construct $\hat{\mathcal{T}}_{i::}$ using the learned low-rank matrices as described in Proposition 3.2.1, which is then treated as a score matrix to predict $\mathcal{T}_{i::}$.

We randomly choose 10% of diseases, genes or chemicals as validation (5%) and test (5%) data and treat the rest 90% as training data for both fiber-wise and slice-wise evaluations. The random sampling procedures are repeated 10 times. The non-negative component matrices of the baseline methods are treated as the CPD-components for constructing the tensor $\hat{\mathcal{T}}$ as what GT-COPR reports. The predictive performances on the test data of all the methods are evaluated by average scores of AUC (area under the receiver operating characteristics), MAP (mean average precision), hits at top 10 (Hits@10) and hits at top 5 (Hits@5), which are summarized in Table 3.2 and 3.3. We can observe that all the three types of product graph regularized GT-COPR have very similar performances in both fiber-wise and slice-wise evaluations; by utilizing the topological information of the product graph via jointly regularizing the tensor elements with the product graph manifolds, GT-COPR consistently and significantly outperforms the other matrix factorization methods in all the fiber-wise evaluations and most of the slice-wise evaluations; SNMF and the soft-weighted methods FASCINATE and wiZAN-Dual also perform clearly better than the binary-weighted methods GWNMTF and GWNMF, which implies that the discrimination between the observed and unobserved bipartite associations are informative, and thus it is more reasonable to consider the unobserved bipartite associations as negative samples than simply treat them as missing entries.

Table 3.2: **Fiber-wise evaluation.**

	Evaluation by gene fibers				Evaluation by disease fibers				Evaluation by chemical fibers			
Methods	AUC	MAP	Hits@10	Hits@5	AUC	MAP	Hits@10	Hits@5	AUC	MAP	Hits@10	Hits@5
GT-COPR (Cartesian)	0.9129	0.1878	0.3440	0.4161	**0.8741**	**0.3006**	0.4337	**0.5630**	0.9749	0.2765	0.3651	**0.4478**
GT-COPR (Tensor)	**0.9132**	**0.1928**	**0.3605**	0.4027	0.8692	0.2842	0.4329	0.5460	**0.9759**	**0.2797**	**0.3827**	0.4399
GT-COPR (Strong)	0.9130	0.1894	0.3468	**0.4192**	0.8697	0.2829	**0.4344**	0.5489	0.9750	0.2759	0.3562	0.4463
SNMF	0.8660	0.1604	0.2827	0.2864	0.7467	0.1463	0.2102	0.2481	0.9236	0.1097	0.1563	0.1877
FASCINATE	0.8978	0.1414	0.2579	0.2522	0.8378	0.2209	0.3310	0.3483	0.9704	0.1489	0.1535	0.1453
wiZAN-Dual	0.7832	0.1287	0.2845	0.2806	0.8678	0.2899	0.4109	0.5495	0.9060	0.1579	0.2651	0.3116
GWNMTF	0.8749	0.0524	0.0185	0.0148	0.7373	0.0886	0.1909	0.1948	0.7076	0.0185	0.0388	0.0519
GWNMF	0.8924	0.0604	0.0753	0.0321	0.7359	0.0597	0.0876	0.0072	0.7241	0.0131	0.0081	0.0028

Table 3.3: **Slice-wise evaluation.**

	Evaluation by gene slices				Evaluation by disease slices				Evaluation by chemical slices			
Methods	AUC	MAP	Hits@10	Hits@5	AUC	MAP	Hits@10	Hits@5	AUC	MAP	Hits@10	Hits@5
GT-COPR (Cartesian)	0.9934	0.0755	0.4776	0.6324	0.9843	0.0302	0.0694	0.0710	0.9825	**0.0463**	**0.2123**	**0.1890**
GT-COPR (Tensor)	**0.9945**	0.0687	**0.5223**	**0.6905**	**0.9853**	**0.0385**	0.0935	0.0903	0.9708	0.0337	**0.2123**	**0.1890**
GT-COPR (Strong)	0.9919	**0.0802**	0.4375	**0.6905**	0.9840	0.0303	0.0694	0.0710	**0.9874**	0.0392	**0.2123**	**0.1890**
SNMF	0.9032	0.0159	0.0759	0.0993	0.8568	0.0152	**0.1403**	**0.1387**	0.9181	0.0241	0.1156	0.1240
FASCINATE	0.9861	0.0223	0.0558	0.0182	0.8698	0.0159	0.0710	0.0613	0.9687	0.0155	0.0532	0.0474
wiZAN-Dual	0.9642	0.0369	0.0339	0.0616	0.9198	0.0111	0.0903	0.1000	0.9435	0.0096	0.1146	0.0994
GWNMTF	0.7624	0.0003	0	0	0.6646	0.0003	0.0032	0.0065	0.8645	0.0005	0	0
GWNMF	0.7583	0.0002	0.0002	0.0005	0.7589	0.0004	0	0	0.8550	0.0003	0	0

3.3.4 Validation on cancer cell line data

In [9], ANOVA analyses of the Genomics of Drug Sensitivity in Cancer (GDSC) cell line data find 182 significant cancer specific interactions between differential drug sensitivity and cancer functional events (CFEs). The analyses are based on 1,250 CFEs including somatic mutations, copy number alterations and DNA hypermethylation; and 265 clinical, clinical developmental and experimental anti-cancer drug compounds. This dataset is considered as an independent source from the integrated dataset described in Section 3.3.1.

We first map the 12 diseases (cancers), 1,250 CEFs and 265 compounds to the ids in our integrated dataset, resulting in 104 interactions among 9 diseases, 670 genes and 100 chemicals. Next, we train each method with its optimal parameter found in the previous section, using all the observed bipartite associations and networks from the integrated database to obtain the low-rank component matrices, which are then used to construct the $9 \times 670 \times 113$ subtensor in $\hat{\mathcal{T}}$. Then we measure the performances of all the methods (GWNMF/GWNMTF are excluded due to their poor performances) by AUC scores for predicting the disease specific *gene-chemical* interactions across all the 9 cancers

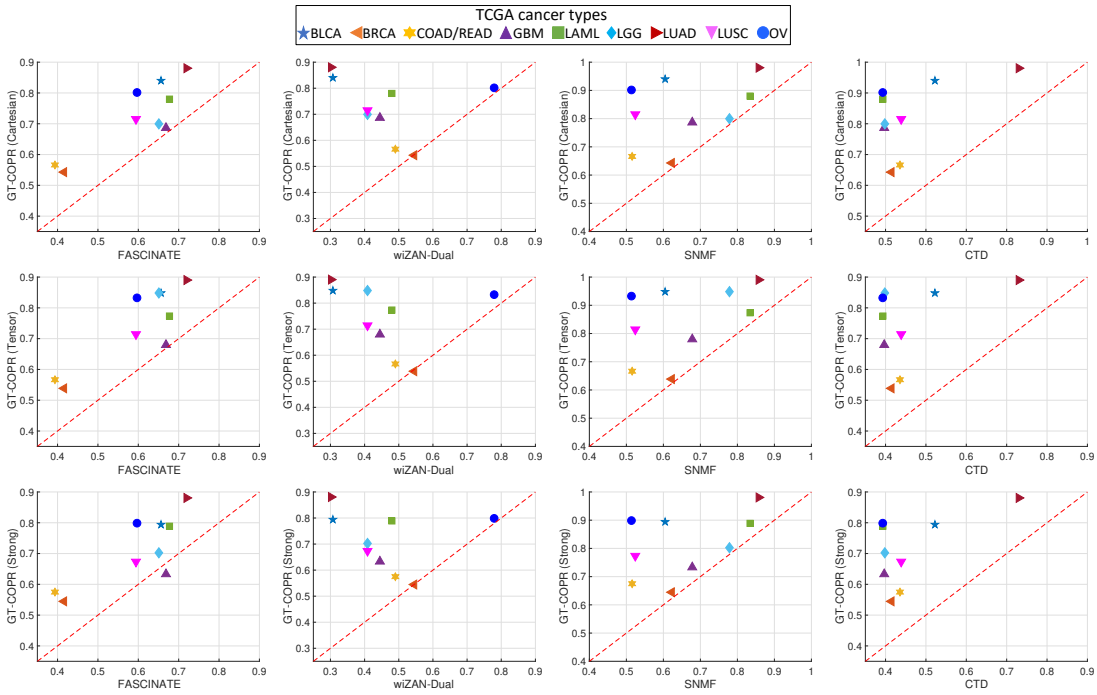


Figure 3.5: **Predicting cancer-specific pharmacogenomic interactions.** The performances of predicting the significant cancer-specific *gene-chemical* interactions in 9 cancer types derived from the cell line dataset are measured by AUC scores.

(correspond to 9 tensor slices). To understand if the predictions made by GT-COPR are biased towards the observed bipartite associations, we also add a baseline using the binary tensor $\mathcal{T}$ constructed from the CTD bipartite associations mentioned in Section 3.3.3 to make the same predictions. The scatter plots in Figure 3.5 show that GT-COPR clearly outperforms the other methods in all the 9 cancer types, which provides the evidence of GT-COPR learning clinically meaningful pharmacogenomic multi-way associations. The right-most column in Figure 3.5 shows that $\mathcal{T}$ has a poor prediction performance on all the cancer types except the Lung Adenocarcinoma (LUAD), which means the CTD database covers only few interactions reported by [9], and further implies the knowledge graphs carry plenty of information for the multi-relational inference. One can also observe that the performances of using the three types of product graph regularization are very close in most of the cancer types. One remarkable difference is

that when predicting the *gene-chemical* interactions in LGG (low grade glioma), the AUC score is lifted by $\backsim$15% via using tensor product graph regularization.

3.3.5 Statistical component analysis

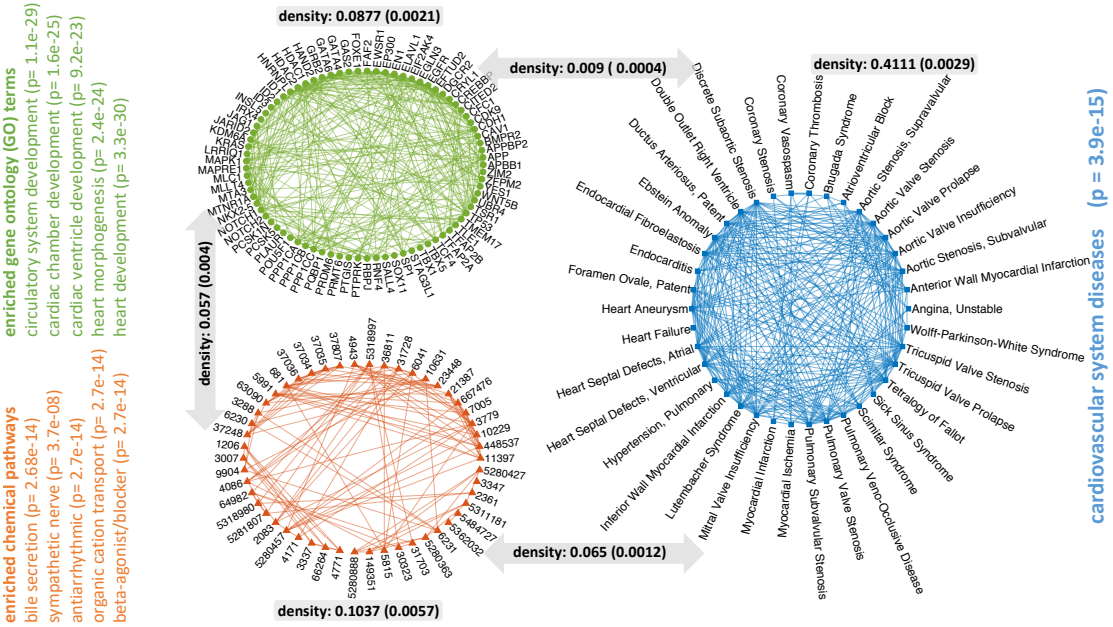


Figure 3.6: **Subnetworks of the detected components.**
The detected components in the cardiovascular system disease category are visualized with subnetworks. The densities of each subnetwork and bipartite subnetwork are given together with the background densities in parentheses.

To evaluate the capability of CT-COPR to find topologically and biologically relevant disease, gene and chemical components, we 1) convert every column (component) in the rank-200 component matrices $A^{(1)}$ (disease), $A^{(2)}$ (gene) and $A^{(3)}$ (chemical) learned by GT-COPR with the Cartesian product graph regularization to z-score vector, 2) select diseases, genes and chemicals with z-scores > 2.33 (p-value < 0.01) matched by components, 3) perform right-tailed Fisher's exact test to find the most significant (p-value $< 10^{-3}$) disease components via comparing with the 13 disease categories reported

Table 3.4: **Detected components of cellular proliferation.**

Disease names	Enriched GO terms	Enriched chemical pathways
breast neoplasms hepatocellular carcinoma adenocarcinoma squamous cell carcinoma colonic neoplasms liver neoplasms lung neoplasms lymphoma cholangiocarcinoma	small molecule metabolic process ($p = 3.1 \times 10^{-21}$) lipid metabolic process ($p = 1.6 \times 10^{-18}$) cellular lipid metabolic process ($p = 1.3 \times 10^{-18}$) metabolic process ($p = 5.9 \times 10^{-18}$) carboxylic acid metabolic process ($p = 1.9 \times 10^{-17}$) oxoacid metabolic process ($p = 2.8 \times 10^{-17}$) organic acid metabolic process ($p = 5.3 \times 10^{-17}$) organic substance metabolic process ($p = 1.67 \times 10^{-15}$)	pathways in cancer (KEGG) ($p = 9.1 \times 10^{-10}$) xenobiotics (Reactome) ($p = 2.4 \times 10^{-8}$) cytochrome P450 (Reactome) ($p = 2.0 \times 10^{-7}$) biological oxidation (Reactome) ($p = 3.8 \times 10^{-7}$) prostate cancer (KEGG) ($p = 4.7 \times 10^{-7}$) chemical carcinogenesis (KEGG) ($p = 1.6 \times 10^{-6}$) breast cancer pathway (Wikipathways) ($p = 6.3 \times 10^{-6}$) hepatocellular carcinoma (KEGG) ($p = 3.4 \times 10^{-5}$)

by [4], and 4) apply gene and chemical enrichment analyses with Gene Ontology (GO) Consortium [50] and IMPaLA [51] respectively to find the GO terms and pathways related to the selected genes and chemicals in each of the disease components found in step 3).

There are 101 biologically related disease gene and chemical components found by the procedures described above. Due to the page limit, we only show the two components corresponding to cardiovascular system and cellular proliferation disease categories respectively. Figure 3.6 shows the top-5 enriched GO terms and chemical pathways, and three subnetworks containing subsets of the matched diseases, genes and chemicals related to the cardiovascular system disease category. Interestingly, the significantly enriched GO terms and chemical pathways are all closely related to the cardiovascular system. For example the heart failure is known to be a syndrome characterized by up regulation of the "sympathetic nervous" system [52]; "organic cation transporters" (OCT1-3 and OCTN1/2) facilitate cardiac uptake of endogenous compounds and numerous drugs [53]; and "antiarrhythmic", "cardiac" and "heart" are all relevant key words. Moreover, each of the three subnetworks is fully connected, with densities of the networks and cross-network bipartite relational matrices significantly higher than the background densities of the integrated dataset given in Figure 3.3. We also show the subsets of the detected diseases in cellular proliferation component, together with two lists of top enriched GO terms and chemical pathways in Table 3.4. The first column shows that all the diseases are neoplasms (caused by an abnormal proliferation of tissues). The last two columns show that the enriched GO terms and chemical pathways are all cancer related. For example, the GO terms "lipid metabolic process" and "cellular lipid metabolic process" are believed to be cancer-development related by a

recent study [54]; the "cytochrome P450" enzymes are known to be important targets in cancer, due to their role in "xenobiotic metabolism" [55]; and chemical pathways of four different types of cancer are also enriched. Overall, the results demonstrate that the components produced by GT-COPR have both topologically and biologically interpretations.

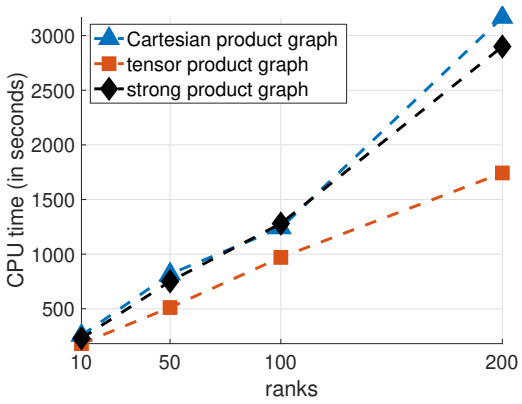


Figure 3.7: **Running time of GT-COPR.**
Comparison of the running time of GT-COPR with three types of product graph regularization.

3.3.6 Implementation and running time

We implemented GT-COPR using MATLAB (R2018a) with the Tensor Toolbox v 2.6 [56] on a server with Intel(R) Xeon(R) CPU E5-2450 (32 cores 2.10GHz, 2 CPUs) and 196GB of RAM. Figure 3.7 shows that GT-COPR is able to learn the *chemical-gene-disease* multi-way associations using the integrated dataset in less than one CPU hour. Empirically, the implementation of the tensor product graph regularized GT-COPR scales almost linearly with the tensor rank and is faster than the Cartesian and strong product graph regularized versions.

3.4 Discussion

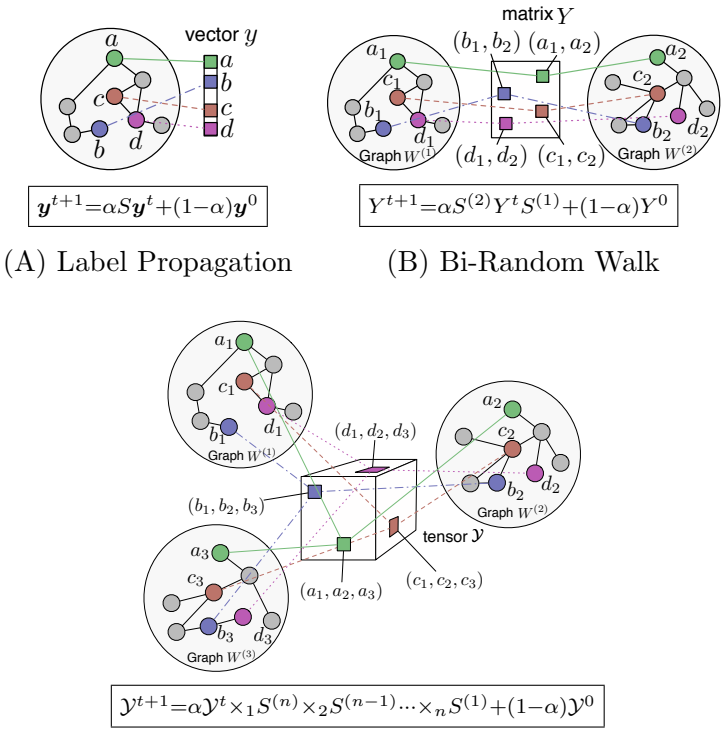
In this chapter, we introduced a novel and general tensor-based algorithm GT-COPR for learning pharmacogenomic multi-way associations across multiple networks, utilizing the observed bipartite associations and the topological information of three different types of product graphs, without relying on known initial multi-way associations. The theoretical analysis of the convergence of GT-COPR will be presented in Section 6.2. We observed that GT-COPR significantly outperforms matrix factorization based methods on predicting the *disease-gene-chemical* multi-way associations on our integrated biomedical dataset. The validation using cancer cell line dataset demonstrates the clinical value of CT-COPR. The statistical component analysis shows that GT-COPR is also able to produce the topologically and biologically relevant disease, gene and chemical components.

Chapter 4

Multi-relational Learning through Label Propagation

4.1 Introduction

Label propagation has been widely used for semi-supervised learning on the similarity graph of labeled and unlabeled samples [28, 35, 36]. As illustrated in Figure 4.1(A), label propagation propagates training labels on a graph S to learn a vector $\boldsymbol{y}$ predicting the labels of nodes. In each iteration of label propagation, each node in the graph receives label information propagated from its neighbors, and also preserves its initial labeling. A generalization of label propagation to the tensor product of two graphs $S^{(1)} \otimes S^{(2)}$, also known as bi-random walk [17], can infer the bipartite associations in a matrix Y between the nodes from the two graphs. The bi-random walk iteration has its matrix form as shown in Figure 4.1(B), smoothing both dimensions of matrix Y with the manifolds of the graphs $S^{(1)}$ and $S^{(2)}$. This approach has been widely adopted in many machine learning applications including biomedical network alignment [17], multi-label classification [57], link prediction [26], collaborative filtering [58], cross-lingual text classification [59], and image segmentation [60]. When generalized to the tensor product of n graphs $\otimes_{i=1}^{n} S^{(i)}$ to predict the associations across the n graphs in an n-way tensor $\mathcal{Y}$ as shown in Figure 4.1(C), label propagation accomplishes n-way relational learning from knowledge graphs [25, 27, 61]. Label propagation on the tensor product graph explores the graph topology for associating nodes across the graphs assuming the



(A) Label Propagation (B) Bi-Random Walk

(C) n-way Relational Learning (illustrated by $n = 3$)

Figure 4.1: **Label propagation generalized to tensor product graphs.** **(A)** Label propagation on a graph predicts the labels of the nodes for semi-supervised learning; **(B)** Label propagation on the tensor product of two graphs predicts links between the nodes across the two graphs; **(C)** Label propagation on an n-way tensor product graph learns n-way multi-way associations across the nodes in n graphs. Each n-way tuple of graph nodes in the same color is represented as an entry in the n-way tensor.

global relations among the nodes reveal the node identities [17]. However, the tensor formulation of label propagation is computationally intensive to solve. Empirically, most of the existing methods are only scalable to learn 3-way associations in large graphs even if the graphs are sparse [25, 27].

The main objective of this chapter is to provide a theoretically justified approximation, and scalable algorithms and implementations to tackle the scalability issue. Our contributions in this study are summarized as follows:

- We propose a novel optimization formulation to approximate the transformation

matrix in the closed-form solution of label propagation on the tensor product graph (TPG), by learning with a subset of eigen-pairs from the normalized TPG. In chapter 6, we will provide the theoretical justification that the globally optimal solution of the optimization problem minimizes an estimating error bound of recovering the true tensor that is structured by the TPG manifolds, for multiple graph alignment; we will also provide a data-dependent error bound using the *transductive Rademacher complexity* for binary hyperlink prediction.

- We develop an efficient eigenvalue selection algorithm to sequentially select the eigen-pairs from each individual normalized graph considering the global spectrum of the transformation matrix. We then show that the spectrum of the low-rank normalized tensor product graph constructed by the selected eigen-pairs is guaranteed to be the globally optimal solution to the proposed optimization formulation.

- We propose the Low-Rank Tensor-based Label Propagation algorithm (LowrankTLP) using efficient tensor operations to compute the approximated solution based on the eigen-pairs selected from the knowledge graphs. We provide an efficient parallel implementation of LowrankTLP using SPLATT library [62] with shared-memory parallelism to increase the scalability by a large magnitude.

- We validate the effectiveness, efficiency and scalability of LowrankTLP on the simulation data by controlling the graph size and topology. We also demonstrate the practical use of LowrankTLP on three real datasets for hyperlink prediction and multiple graph alignment, across a large number of knowledge graphs.

4.2 Problem formulation

Now, we introduce the two multi-relational learning problems studied in this chapter. The objective is to score the queried n-way associations among the nodes across multiple undirected graphs $\{G^{(i)} = (V^{(i)}, E^{(i)}) : i = 1, \ldots, n\}$ where each graph $G^{(i)}$ has $|V^{(i)}| = I_i$ nodes, for either hyperlink prediction or multiple graph alignment given the input as 1) the labels of a small number of observed n-way relations, 2) or the similarity scores between the nodes of every pair of graphs respectively, as illustrated in Figure

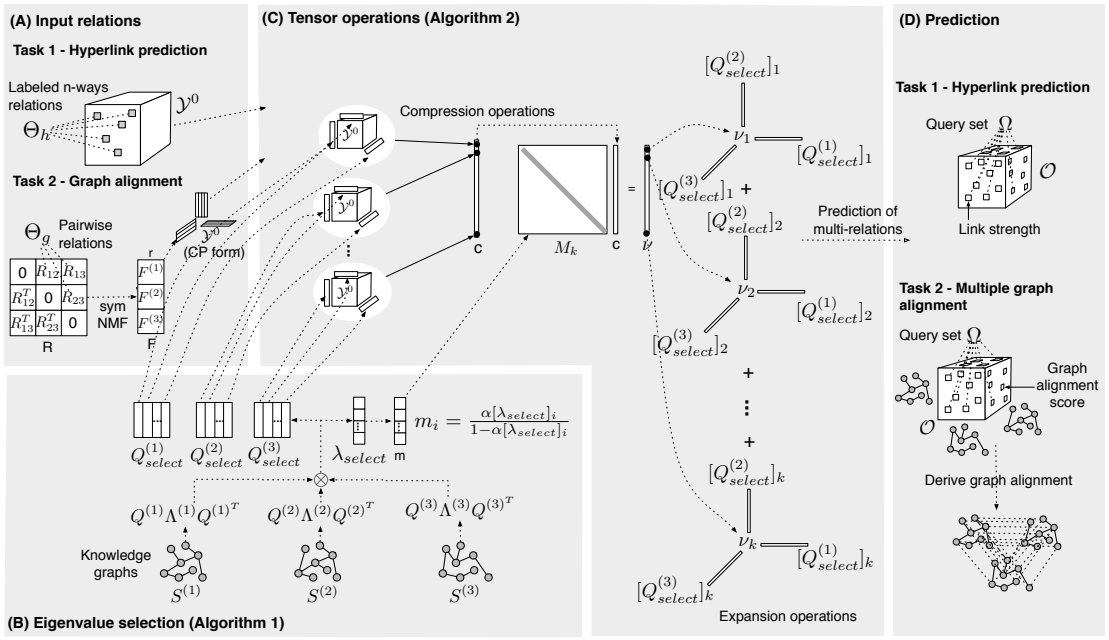


Figure 4.2: **Overview of LowrankTLP algorithm.**
(A) Input: the initial input tensor $\mathcal{Y}^0$ is 1) a sparse tensor of labeled multi-way associations (given in set Θ_h) for hyperlink prediction or 2) a CPD-form estimated from bipartite similarities (given in set Θ_g) between every pair of the graphs for multiple graph alignment. **(B) Knowledge Graphs:** Three normalized undirected graphs $S^{(1)}$, $S^{(2)}$ and $S^{(3)}$ are given. Algorithm 4.1 will obtain the k optimal eigenvalues and the corresponding eigenvectors of the tensor product graph based on the eigen-decomposition of each graph for computing the approximation of the closed-form solution. **(C) Efficient Tensor Operations:** $\mathcal{Y}^0$ and the selected eigen-pairs are used to perform compression and prediction operations to obtain the approximated closed-form solution of label propagation in a CPD-form. **(D) Output:** The scores of the multi-way associations queried in set Ω are predicted in a sparse tensor $\mathcal{O}$ to represent the hyperlink strengths, or to derive the alignment of multiple graphs.

4.2(A). These two learning tasks correspond to the two input options discussed in Section 1.2.

Task 1: hyperlink prediction

- *Input associations* (Option 1 in Section 1.2): a small set $\Theta_h = \{(i_1, i_2, \ldots, i_n) : i_j \in [1, I_j], \forall j = 1, \ldots, n\}$ of labeled n-way associations (hyperlinks) among the nodes across n graphs, where i_j denotes the node of graph $G^{(j)}$.

- *Queried multi-way associations*: a set $\Omega = \{(j_1, j_2, \ldots, j_n) : j_l \in [1, I_l], \forall l = 1, \ldots, n\}$ of the queried n-way associations (hyperlinks), chosen per user's interests.

- *Learning task*: given n knowledge graphs and the set Θ_h of labeled hyperlinks, predict the link strengths of the queried set Ω of hyperlinks in a sparse tensor $\mathcal{O} \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ with $|\Omega|$ nonzero entries.

Task 2: multiple graph alignment

- *Input associations* (Option 2 in Section 1.2): a set $\Theta_g = \{R_{i,j} \in \mathbb{R}_+^{I_i \times I_j} : \forall i, j \in [1, n] \text{ and } i < j\}$ of non-negative matrices, where $R_{i,j}$ contains the similarity scores between the nodes of a pair of graphs $G^{(i)}$ and $G^{(j)}$.

- *Queried multi-way associations*: a set $\Omega = \{(j_1, j_2, \ldots, j_n) : j_l \in [1, I_l], \forall l = 1, \ldots, n\}$ of the queried n-way associations, which can be derived from the bipartite associations by heuristic as in [63] and [64].

- *Learning task*: given n knowledge graphs and the set Θ_g of bipartite associations, predict the alignment scores between the queried set Ω of n-way tuples of graph nodes in a sparse tensor $\mathcal{O} \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ with $|\Omega|$ nonzero entries.

In both tasks described above, we expect to learn a sparse tensor $\mathcal{O}$ which stores the scores to return for any queried n-way relations. Given either the labels of the observed n-way relations or the similarity scores between the nodes of all the graph pairs, we will show that tensor $\mathcal{O}$ can be learned from a compressed tensor $\mathcal{Y} \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ using label propagation on the normalized tensor product graph $S = \otimes_{i=1}^{n} S^{(i)}$ which is

discussed in Section 2.4. We propose to model both learning learning tasks with the regularization framework presented in Equation (2.3) and explained below,

$$\mathcal{J}(\mathcal{Y}) = \frac{1}{2}(\mathbf{vec}(\mathcal{Y})^T(I - \otimes_{i=1}^{n} S^{(i)})\mathbf{vec}(\mathcal{Y}) + \mu||\mathbf{vec}(\mathcal{Y}) - \mathbf{vec}(\mathcal{Y}^0)||_2^2).$$

- ***Formulation of hyperlink prediction (Task 1)***: the learning task 1 is a *transductive learning* problem [65, 66] of inferring a tensor $\mathcal{Y}$ from a sparse initial tensor $\mathcal{Y}^0$. The nonzero entries in $\mathcal{Y}^0$ are the labels (link types) of the observed n-way associations (hyperlinks) given in set Θ_h (defined in Section 4.2), such that the label of the $(i_1, i_2, \ldots, i_n)$-th hyperlink is $\mathcal{Y}^0_{i_n i_{n-1} \ldots i_1}$. The zero entries in $\mathcal{Y}^0$ represent the unobserved n-way associations. The inferred tensor $\mathcal{Y}$ is composed of the link strengths of all the n-way hyperlinks.

- ***Formulation of multiple graph alignment (Task 2)***: we convert the set Θ_g (defined in Section 4.2) of bipartite similarity matrices to the rank-r CPD-form of $\mathcal{Y}^0$ as following: first, symmetric NMF (symNMF) [49] is applied on a symmetric matrix R built by stacking all $R_{i,j}$'s to obtain a nonnegative component matrix $F \in \mathbb{R}_+^{(\sum_{i=1}^{n} I_i) \times r}$ such that $R \approx FF^T$ as illustrated in Figure 4.2 (A). Then, the rank-r CPD-form of $\mathcal{Y}^0$ is approximated as $\mathcal{Y}^0 = [\![F^{(n)}, F^{(n-1)}, \ldots, F^{(1)}]\!]$ where $F^{(i)} \in \mathbb{R}_+^{I_i \times r}$ is the i-th submatrix of F. The assumption is that more similar bipartite associations between every pair $(i_a, i_b) \subset (i_1, i_2, \ldots, i_n)$ imply a stronger n-way associations in tuple $(i_1, i_2, \ldots, i_n)$. This representation has been widely adopted in real graph alignment problems as in [63], [10] and [67]. As $\mathcal{Y}^0$ is guessed from the bipartite associations, we call the learning task 2 *structured signal recovery from noisy observation*. Our goal is to recover the true tensor $\mathcal{Y}$ which is structured by the TPG manifolds, from its noisy observation $\mathcal{Y}^0$.

As pointed out in Section 2.5, the time and space complexities of computing label propagation are $O(\prod_{i=1}^{n} I_i)$ and $O((\prod_{i=1}^{n} I_i)(\sum_{i=1}^{n} I_i))$ respectively. Therefore, it is challenging to directly apply label propagation on a high-order tensor. To tackle this challenge, we propose the LowrankTLP algorithm in Section 4.3 based on a principled approximation of the linear transformation matrix $(I - \alpha S)^{-1}$ in the closed-form solution in Equation (2.8). We also outline the framework of LowrankTLP in Figure 4.2.

4.3 Low-rank label propagation

In this section, we propose the scalable LowrankTLP algorithm based on a theoretically principled approximation to the closed-form solution given in Equation (2.8). Section 4.3.1 proposes the optimization formulation for approximating the closed-form solution. The globally optimal solution can be obtained by selecting a subset of eigen-pairs from the normalized TPG S using Algorithm 4.1 presented in Section 4.3.2. The scalable LowrankTLP algorithm is then proposed in Section 4.3.3, using the eigen-pairs selected by Algorithm 4.1. Section 4.3.4 analyzes the time and space complexity of LowrankTLP. In Chapter 6, we will provide the theoretical justification of our optimization formulation proposed in Section 4.3.1 by providing an estimating error bound for recovering the true tensor that is structured by the TPG manifolds, and a data-dependent error bound for justifying LowrankTLP in a special case of *Task 1*.

4.3.1 Optimization formulation

We propose to approximate $\mathcal{Y}^*$ in Equation (2.8), which is the closed-from solution to the objective $\mathcal{J}(\mathcal{Y})$ in Equation (2.3), by minimizing the perturbation on transformation matrix $(I - \alpha S)^{-1}$ as follows,

$$\begin{aligned} \underset{eig(S_k)}{\text{minimize}} \quad & ||(I - \alpha S)^{-1} - (I - \alpha S_k)^{-1}||_{2,F} \\ \text{subject to} \quad & \text{rank}(S_k) = k,\ eig(S_k) \subseteq eig(S), \end{aligned} \tag{4.1}$$

where $S = \otimes_{i=1}^{n} S^{(i)}$ is the normalized TPG; $eig(S_k)$ and $eig(S)$ denote the sets of eigen-pairs of S_k and S respectively; $||.||_2$ is spectral norm and $||.||_F$ is Frobenius norm.

The objective is to find a low-rank matrix S_k defined by a subset of eigen-pairs of S to give the lowest divergence on the overall transformation matrix $(I - \alpha S)^{-1}$. The selected eigen-pairs in $eig(S_k)$ will be used for constructing the approximated closed-form solution $\widehat{\mathcal{Y}}^*$ in Equation (4.5) in Section 4.3.2, which forms the foundation of LowrankTLP (Algorithm 4.2) described in Section 4.3.3.

Later, in Section 6.1.1, we will show this formulation minimizes the estimating error bound in Theorem 6.1.1. It is noteworthy that simply computing the best rank-k approximation to S per Eckart-Young-Mirsky theorem does not guarantee the optimal

solution. Instead, we will show that the globally optimal solution to the optimization problem (4.1) can be efficiently found by Algorithm 4.1.

Algorithm 4.1: Select Eigenvalues

Data: $\{S^{(i)} : i = 1, \ldots, n\}$, $\alpha \in (0, 1)$.

Result: $\boldsymbol{\lambda}_{\text{select}}$ and $\{Q^{(i)}_{\text{select}} : i = 1, \ldots, n\}$.

Compute and store the eigenvalues and eigenvectors of $S^{(i)}$ in vector $\boldsymbol{\lambda}^{(i)}$ and matrix $Q^{(i)}$ respectively, for $i = 1, \ldots, n$.

$\Gamma \leftarrow \boldsymbol{\lambda}^{(1)}$

for $i = 2$ *to* n **do**

  $\Gamma \leftarrow \boldsymbol{\lambda}^{(i)} \otimes \textbf{top\_bot\_2k}(\Gamma)$

end

$\boldsymbol{\lambda}_{\text{select}} \leftarrow k$ elements from Γ with the largest $\frac{\alpha|\Gamma_j|}{1-\alpha\Gamma_j}, j = 1, \ldots, k$

for $i = n$ *to* 1 **do**

  Return $Q^{(i)}_{\text{select}}$ from $Q^{(i)}$ by looking-up indexes of the entries in Γ.

end

4.3.2 Selection of the optimal eigen-pairs

Let $S_k = Q_{1:k}\Lambda_{1:k}Q_{1:k}^T$ be the eigen-decomposition of S_k, where $\Lambda_{1:k}$ and $Q_{1:k}$ store the eigen-pairs $\{(\lambda_j, \boldsymbol{q}_j) : j = 1, \ldots, k\}$ of S_k selected from $eig(S)$. Also, define diagonal matrix Λ_{rest} and matrix Q_{rest} to hold the remaining eigen-pairs $\{(\lambda_i, \boldsymbol{q}_i) : i = k + 1, \ldots, N\}$ of $eig(S)$, where $N = \prod_{l=1}^{n} I_l$. According to Lemma A.2.3 in appendix, we have

$$\lambda_j = \prod_{i=1}^{n} \lambda_j^{(i)} \quad \text{and} \quad \boldsymbol{q}_j = \otimes_{i=1}^{n} \boldsymbol{q}_j^{(i)}, \forall j = 1, \ldots, k,$$

where $\lambda_j^{(i)}$ and $\boldsymbol{q}_j^{(i)}$ is an eigen-pair of $S^{(i)}$ contributing to λ_j and $\boldsymbol{q}_j$ of S. This implies the eigen-pairs of S_k are composed of the properly selected eigen-pairs from each $S^{(i)}$, for $i = 1, \ldots, n$.

Proposition 4.3.1. *Define $A = (I - \alpha S)^{-1}$ and its approximation $\hat{A} = (I - \alpha S_k)^{-1}$. According to Woodbury formula [68], we have*

$$\hat{A} = Q_{1:k}((I - \alpha\Lambda_{1:k})^{-1} - I)Q_{1:k}^T + I. \tag{4.2}$$

Theorem 4.3.1. *The optimal k eigenvalues $\{\lambda_j : j = 1, \ldots, k\}$ that solve the optimization problem* (4.1) *are among the union of the k largest (algebraic) and k smallest (algebraic) eigenvalues of S and satisfy the following condition*

$$\frac{\alpha|\lambda_j|}{1-\alpha\lambda_j} \geq \frac{\alpha|\lambda_i|}{1-\alpha\lambda_i}, \forall j \in [1,k], \forall i \in [k+1, N].$$

Proof. Given Equation (4.2), the perturbation is obtained as

$$\hat{A} - A = Q_{\text{rest}}(I - (I - \alpha\Lambda_{\text{rest}})^{-1})Q_{\text{rest}}^T,$$

whose singular values are $\{\frac{\alpha|\lambda_i|}{1-\alpha\lambda_i} : i = k+1, \ldots, N\}$ and k zeros. Thus, its spectral norm and Frobenius norm are

$$\begin{aligned} ||\hat{A} - A||_2 &= \frac{\alpha|\lambda^*|}{1-\alpha\lambda^*} \text{ and} \\ ||\hat{A} - A||_F &= \sqrt{\sum_{i=k+1}^{N} (\frac{\alpha|\lambda_i|}{1-\alpha\lambda_i})^2}, \end{aligned} \tag{4.3}$$

where $\lambda^* = \text{argmax}_{\lambda \in \{\lambda_{k+1}, \ldots, \lambda_N\}} \frac{\alpha|\lambda|}{1-\alpha\lambda}$. To minimize both norms in Equation (4.3), the k selected eigenvalues $\{\lambda_j : j = 1, \ldots, k\}$ should produce the largest elements in the set $\{\frac{\alpha|\lambda_j|}{1-\alpha\lambda_j} : j = 1, \ldots, k\}$ among all the eigenvalues of S. In addition, since $\alpha \in (0, 1)$ and $\lambda_j \in [-1, 1]$ for $j = 1, \ldots, k$, the function $\frac{\alpha|\lambda_j|}{1-\alpha\lambda_j}$ is monotonically increasing in the positive orthant and decreasing in the negative orthant with λ_j. Thus, $\{\lambda_j : j = 1, \ldots, k\}$ must be in the union of the k largest (algebraic) eigenvalues and k smallest (algebraic) eigenvalues of S. (End of Proof) □

Algorithm 4.2: LowrankTLP

Data: $\{S^{(i)} : i = 1, \ldots, n\}$, $\mathcal{Y}^0$, α, k and Ω.

Result: Sparse tensor O.

Apply Algorithm 4.1 to obtain $\boldsymbol{\lambda}_{\text{select}}$, $\{Q^{(i)}_{\text{select}} : i = 1, \ldots, n\}$.
Initialize $\boldsymbol{v}$ to be a k-D vector with all zeros.
if $\mathcal{Y}^0$ *is sparse* **then**
  for *j=1 to k* **do**
    $\boldsymbol{v}_j \leftarrow \mathcal{Y}^0 \bar{\times}_1 \boldsymbol{q}_j^{(n)} \bar{\times}_2 \boldsymbol{q}_j^{(n-1)} \ldots \bar{\times}_n \boldsymbol{q}_j^{(1)}$
  end
end
if $\mathcal{Y}^0$ *is in CPD-form* $[\![F^{(n)}, F^{(n-1)}, \ldots, F^{(1)}]\!]$ **then**
  $\Psi \leftarrow Q^{(1)T}_{select} F^{(1)}$
  for $j = 2$ *to* k **do**
    $\Psi \leftarrow \Psi \circledast (Q^{(j)T}_{select} F^{(j)})$
  end
  $\boldsymbol{v} \leftarrow \Psi \mathbf{1}$
end
$m \leftarrow \alpha \boldsymbol{\lambda}_{\text{select}} / (1 - \alpha \boldsymbol{\lambda}_{\text{select}})$
$\hat{\boldsymbol{v}}' \leftarrow (\boldsymbol{v} \circledast m)'$
Initialize $\mathcal{O} = \{\}$ to be an empty tensor.
for *every tuple* $(i_1, i_2, \ldots, i_n)$ *in* Ω **do**
  $\mathcal{O}_{i_n i_{n-1} \ldots i_1} \leftarrow (1 - \alpha)(\sum_{j=1}^{k} \hat{\boldsymbol{v}}'_{\boldsymbol{j}} \prod_{l=1}^{n} q^{(l)}_{i_l,k} + \mathcal{Y}^0_{i_n i_{n-1} \ldots i_1})$
end

Theorem 4.3.2. *Define function* $\boldsymbol{top\_bot\_2k}(\boldsymbol{x}) = \boldsymbol{top\_k}(\boldsymbol{x}) \cup \boldsymbol{bot\_k}(\boldsymbol{x})$ *where* $\boldsymbol{top\_k}(\boldsymbol{x})$ *and* $\boldsymbol{bot\_k}(\boldsymbol{x})$ *return the k algebraically largest and smallest values of the vector* $\boldsymbol{x}$ *respectively. Given the vector* $\boldsymbol{\lambda}^{(i)}$ *of the eigenvalues of* $S^{(i)}$ *for* $i = 1, \ldots, n$*, we have*

$$\boldsymbol{top\_bot\_2k}(\otimes_{i=1}^{n} \boldsymbol{\lambda}^{(i)}) = \boldsymbol{top\_bot\_2k}(\boldsymbol{\lambda}^{(n)} \otimes \boldsymbol{top\_bot\_2k}(\Gamma^{(n-1)})), \text{ where}$$

$$\Gamma^{(i)} = \begin{cases} \boldsymbol{\lambda}^{(i)} \otimes \boldsymbol{top\_bot\_2k}(\Gamma^{(i-1)}), & \text{if } i = 2, \ldots, n-1 \\ \boldsymbol{\lambda}^{(1)}, & \text{if } i = 1. \end{cases}$$

Proof. Theorem 4.3.2 can be proven by induction based on the observation that the k

algebraically largest (smallest) elements in the outer product of two real vectors can only be among the multiplications between the union of the k largest and smallest values in the two vectors. Thus, only the numbers in $\mathbf{top\_bot\_2k}(\Gamma^{(i-1)})$ are needed to compute the next $\Gamma^{(i)}$ in the recursion. Taking the elements in $\mathbf{top\_bot\_2k}(\Gamma^{(i-1)})$ in the multiplication with each $\boldsymbol{\lambda}^{(i)}$ guarantees that the numbers needed for computing the k largest (smallest) elements in $\otimes_{i=1}^{n}\boldsymbol{\lambda}^{(i)}$ will be kept in $\Gamma^{(i)}$. (End of Proof) $\square$

According to Theorem 4.3.1, the selected eigenvalues $\{\lambda_j : j = 1, \ldots, k\}$ from S satisfying $\frac{\alpha|\lambda_j|}{1-\alpha\lambda_j} \geq \frac{\alpha|\lambda_i|}{1-\alpha\lambda_i}, \forall j \in [1, k], \forall i \in [k+1, N]$ must be contained in the union of the k largest and k smallest eigenvalues of S. Thus, we only need to find the $\mathbf{top\_bot\_2k}(\otimes_{i=1}^{n}\boldsymbol{\lambda}^{(i)})$ with Theorem 4.3.2, and select k eigenvalues which give the largest elements in the set $\{\frac{\alpha|\lambda_j|}{1-\alpha\lambda_j} : j = 1, \ldots, k\}$. Based on the idea, we propose Algorithm 4.1 to select the eigen-pairs $\{(\lambda_j^{(i)}, \boldsymbol{q}_j^{(i)}) : i = 1, \ldots, n, j = 1, \ldots, k\}$ efficiently in time $O(\sum_{i=1}^{n}(kI_i \log(kI_i))$, plus the time for eigen-decomposition of each knowledge graph. Algorithm 4.1 starts with $\boldsymbol{\lambda}^{(1)}$, the eigenvalues of the first graph (line 2) and iteratively merges another $\boldsymbol{\lambda}^{(i)}$ one at a time in the for-loop between line 3-5 to compute $\mathbf{top\_bot\_2k}(\otimes_{l=1}^{i}\boldsymbol{\lambda}^{(l)})$ in Γ. Each merge step computes and sorts $O(kI_i)$ numbers. Algorithm 4.1 outputs a vector $\boldsymbol{\lambda}_{\text{select}}$ of the selected eigenvalues from S and matrices $Q_{\text{select}}^{(i)}$ of the selected eigenvectors from $Q^{(i)}$, for $i = 1, \ldots, n$ such that

$$\begin{aligned}\boldsymbol{\lambda}_{\text{select}} &= [\lambda_1, \lambda_2, \ldots, \lambda_k]^T \\ Q_{\text{select}}^{(i)} &= [\boldsymbol{q}_1^{(i)}, \boldsymbol{q}_2^{(i)}, \ldots, \boldsymbol{q}_k^{(i)}], \forall i = 1, \ldots, n.\end{aligned}$$

Define $M = (I - \alpha\Lambda_{1:k})^{-1} - I$, which is computed from $\boldsymbol{\lambda}_{\text{select}}$ as

$$M = \text{diag}\left(\left[\frac{\alpha\lambda_1}{1-\alpha\lambda_1}, \frac{\alpha\lambda_2}{1-\alpha\lambda_2}, \ldots, \frac{\alpha\lambda_k}{1-\alpha\lambda_k}\right]\right).$$

The matrix $Q_{1:k}$ can be computed from $\{Q_{\text{select}}^{(i)} : i = 1, \ldots, n\}$ as $Q_{1:k} = \odot_{i=1}^{n} Q_{\text{select}}^{(i)}$. By Equation (4.2), the closed-form solution in Equation (2.8) can be approximated as

$$\begin{aligned}\mathbf{vec}(\widehat{\mathcal{Y}}^*) =& (1-\alpha)\hat{A}\mathbf{vec}(\mathcal{Y}^0) && (4.4)\\ =& (1-\alpha)(\odot_{i=1}^{n} Q_{\text{select}}^{(i)})M(\odot_{i=1}^{n} Q_{\text{select}}^{(i)})^T\mathbf{vec}(\mathcal{Y}^0) + (1-\alpha)\mathbf{vec}(\mathcal{Y}^0). && (4.5)\end{aligned}$$

4.3.3 LowrankTLP algorithm

Equation (4.5) implies a 2-step tensor computation of the closed-form solution given in Algorithm 4.2. The two steps are also illustrated in Figure 4.2.

Compression step (line 1-16 in Algorithm 4.2)

- $\mathcal{Y}^0$ ***is sparse in hyperlink prediction (Task 1)*** (line 2-7): the role of $(\odot_{i=1}^{n} Q_{\text{select}}^{(i)})^T \mathbf{vec}(\mathcal{Y}^0)$ in Equation (4.5) is to compress the original tensor $\mathcal{Y}^0$ to a k-D vector $\boldsymbol{v}$ with its jth element

$$\boldsymbol{v}_j = \mathcal{Y}^0 \bar{\times}_1 \boldsymbol{q}_j^{(n)} \bar{\times}_2 \boldsymbol{q}_j^{(n-1)} \bar{\times}_3 \ldots \bar{\times}_n \boldsymbol{q}_j^{(1)}, \tag{4.6}$$

  where each $\bar{\times}_i$ denotes mode-i vector product of tensor. In Equation (4.6), the original tensor $\mathcal{Y}^0$ is compressed to a scalar by multiplying with n vectors which is similar to computing the core tensor in Tucker decomposition. Denote the number of nonzeros in $\mathcal{Y}^0$ as $|\mathcal{Y}^0|$, the time complexity of the compression step is $O(|\mathcal{Y}^0|nk)$.

  $\star$ ***Parallel implementation***: The construction of $\boldsymbol{v}$ via Equation (4.6) performs j sequences of n-way tensor-vector products as

$$\begin{aligned} Z &\leftarrow Y_{(1)}^0 (Q_{\text{select}}^{(2)} \odot \cdots \odot Q_{\text{select}}^{(n)}), \\ \boldsymbol{v}_j &\leftarrow \boldsymbol{q}_j^{(1)T} \boldsymbol{z}_j \qquad \forall j = 1, \ldots, k, \end{aligned} \tag{4.7}$$

  where $Y_{(1)}^0$ denotes the matrix flattened from $\mathcal{Y}^0$. The kernel in Equation (4.7) is similar to *matricized tensor times Khatri-Rao product (MTTKRP)* [69] involving $n-1$ products during the computation of the CPD. Therefore, we can leverage parallel algorithms developed to compute the CPD for the computation. We adopt SPLATT [70], a C library with shared-memory parallelism for fast *MTTKRP* computation. Parallelized in p threads, the parallel implementation reduces the complexity to $O(\frac{|\mathcal{Y}^0|nk}{p})$.

- $\mathcal{Y}^0$ ***is in CPD-form in multiple graph alignment (Task 2)*** (line 8-14): when the initial tensor $\mathcal{Y}^0$ is in the CPD-form $[\![F^{(n)}, F^{(n-1)}, \ldots, F^{(1)}]\!]$ the k-D vector $\boldsymbol{v}$

can be obtained by

$$\boldsymbol{v} = (\odot_{i=1}^{n} Q_{\text{select}}^{(i)})^T (\odot_{i=1}^{n} F^{(i)})\mathbf{1} \tag{4.8}$$

$$= \circledast_{i=1}^{n} (Q_{\text{select}}^{(i)T} F^{(i)})\mathbf{1}, \tag{4.9}$$

where Equation (4.8) is obtained by *vectorization property* of CPD-form (Definition A.1.1 in appendix) and Equation (4.9) can be derived from Lemma A.2.2 in appendix. Since each $Q_{\text{select}}^{(i)T} F^{(i)}$ takes $O(krI_i)$ (recall r is the rank of the $\mathcal{Y}^0$ in CPD-form), the time complexity of the compression step becomes only $O(kr\sum_{i=1}^{n} I_i)$.

Expansion (Prediction) step: (line 17-20 in Algorithm 4.2)
After obtaining the k-D vector $\boldsymbol{v}$ which is then multiplied by the diagonal matrix M to obtain another k-D vector $\hat{\boldsymbol{v}}$, the second step is to compute

$$\mathbf{vec}(\widehat{\mathcal{Y}}^*) = (1-\alpha)((\odot_{i=1}^{n} Q_{\text{select}}^{(i)})\hat{\boldsymbol{v}} + \mathbf{vec}(\mathcal{Y}^0)). \tag{4.10}$$

The left term of (4.10) has the same form as the vectorized CPD with component matrices $Q_{\text{select}}^{(i)} \in \mathbb{R}^{I_i \times k}$ for $i = 1, \ldots, n$. Thus, the tensorized form can be obtained as

$$\widehat{\mathcal{Y}}^* = (1-\alpha)(\llbracket \hat{\boldsymbol{v}}'; Q_{\text{select}}^{(n)}, Q_{\text{select}}^{(n-1)}, \ldots, Q_{\text{select}}^{(1)} \rrbracket + \mathcal{Y}^0), \tag{4.11}$$

where $\hat{\boldsymbol{v}}'$ is a reversal of the elements in $\hat{\boldsymbol{v}}$. According to Equation (4.11), the k-D vector $\hat{\boldsymbol{v}}$ and matrices $\{Q_{\text{select}}^{(i)} : i = 1, \ldots, n\}$ together with $\mathcal{Y}^0$ store all the information for computing any entry of $\widehat{\mathcal{Y}}^*$ with a time complexity $O(nk)$. Suppose the query set Ω has cardinality $|\Omega|$, the total time complexity for predicting the queried n-way associations in a sparse tensor $\mathcal{O}$ is $O(nk|\Omega|)$.

Table 4.1: **Comparison of time complexities.**

	Compression step	Prediction step
LowrankTLP (Task 1)	$O(\sum_{i=1}^{n}(I_i^3 + kI_i\log(kI_i)) + \lvert\mathcal{Y}^0\rvert nk)$	$O(nk\lvert\Omega\rvert)$
LowrankTLP (Task 2)	$O(\sum_{i=1}^{n}(I_i^3 + kI_i\log(kI_i) + krI_i))$	$O(nk\lvert\Omega\rvert)$
ApproxLink (Task 1)	$O(\sum_{i=1}^{n}(I_i k_i^2 + k_i^3) + \lvert\mathcal{Y}^0\rvert n(\prod_{i=1}^{n} k_i))$	$O(n(\prod_i k_i)\lvert\Omega\rvert)$
GraphCP (Task 1)	$O(iters * n(\lvert\mathcal{Y}^0\rvert nr + r^2\sum_{i=1}^{n} I_i + r\sum_{i=1}^{n} I_i^2))$	$O(nr\lvert\Omega\rvert)$
GraphCP (Task 2)	$O(iters * n(r^2\sum_{i=1}^{n} I_i + r\sum_{i=1}^{n} I_i^2))$	$O(nr\lvert\Omega\rvert)$

4.3.4 Time and space complexities

The time complexity of selecting the optimal eigen-pairs using Algorithm 4.1 is $O(\sum_{i=1}^{n} I_i^3 + kI_i \log(kI_i))$. Therefore, the overall complexity of the computing the compressed representation in Algorithm 4.2 (line 1 - 16) is $O(\sum_{i=1}^{n}(I_i^3 + kI_i \log(kI_i)) + |\mathcal{Y}^0|nk)$ for sparse initialization, and $O(\sum_{i=1}^{n}(I_i^3 + kI_i \log(kI_i) + krI_i))$ for CPD-form initialization. The time complexity of the prediction step in Algorithm 4.2 (line 17 - 20) is $O(nk|\Omega|)$ for both sparse initialization and CPD-form initialization. Table 4.1 compares the time complexity of LowrankTLP with existing tensor-based multi-relational learning methods described in Section 4.4.1. Note that compared with LowrankTLP, ApproxLink [26] has slightly lower compression complexity when n is small; however, when n is big the complexity of ApproxLink in terms of $\prod_{i=1}^{n} k_i$, where k_i is the rank of the i-th graph, becomes a bottleneck in the computation. For GraphCP [22], we assume the first order method is applied to minimize the objective functions. Note that the overall complexity of GraphCP is the number of iterations multiplies the per-iteration-complexity (computing the gradient) in the compression step while the empirical runtime relies on the optimization method, line search type, initialization, stopping condition and etc.

The space required to store the eigenvectors of all the normalized graphs is $O(\sum_{i=1}^{n} I_i^2)$; to store the indexes of the selected eigen-pairs is $O(k)$; and to store the initial tensor is $O(|\mathcal{Y}^0|)$ and $O(\sum_{i=1}^{n} I_i r)$ for sparse and CPD-form initial tensor respectively. Thus, the overall space complexity is $O(|\mathcal{Y}^0| + \sum_{i=1}^{n} I_i^2 + k)$ for sparse initialization and $O(\sum_{i=1}^{n} I_i^2 + \sum_{i=1}^{n} I_i r + k)$ for CPD-form initialization.

4.4 Experiments

In the experiments, the performance of LowrankTLP for hyperlink prediction (Task 1) and multiple graph alignment (Task 2) was evaluated in simulation and three real datasets. Section 4.4.1 introduces the baseline methods and provides the details of experimental setups. Section 4.4.2 evaluates the effectiveness and efficiency of LowrankTLP for hyperlink prediction (Task 1) on the simulation data, through controlling the size and topology of multiple artificial graphs. Section 4.4.3 tests the practical application of LowrankTLP for hyperlink prediction (Task 1) on the DBLP dataset of scientific publication records. In Section 4.4.4 and 4.4.5, we evaluate the practical application

Table 4.2: **Summary of datasets in the experiments.**

Task 1: hyperlink prediction			
Experiment	Input associations	Query set	Knowledge graph
Simulation	observed n-way associations	held-out test n-way associations	n graphs generated by permuting a percentage of edges from a common random graph
DBLP	sampled known (author, paper, venue)-associations	held-out known (author,paper,venue)- associations	Author × Author, Paper × Paper and Venue × Venue graphs
Task 2: multiple graph alignment			
Experiment	Input associations	Query set	Knowledge graph
CT scans	RBF similarities across spots sampled from each pair of CT scan images	alignment scores of spots across multiple images	RBF similarities between spots sampled within each CT scan image
PPI	BLAST sequence similarities between proteins from each pair of species	alignment scores of proteins across multiple species	protein-protein interactions (PPI) networks for different species

of LowrankTLP to multiple graph alignment (Task 2). We first apply LowrankTLP to align up to 26 CT scan images in Section 4.4.4. Next, we evaluate the performance of LowrankTLP for the global alignment of up to 4 full protein-protein interaction (PPI) networks across 4 different species in Section 4.4.5. For better clarity, we also summarize the input/output of all the experiments in Table 4.2.

4.4.1 Baseline methods and implementations

Baseline methods

We compared LowrankTLP with seven baseline methods in the simulations and the experiments based on their applicability to hyperlink prediction and multiple graph alignment.

- *Approximate link propagation* (**ApproxLink**) [26]: ApproxLink was originally designed for pair-wise link prediction in a matrix. We extended its operations for hyperlink prediction in a tensor. Given an incomplete initial tensor $\mathcal{Y}^0 \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ with zeros representing the missing entries, and n knowledge graphs $\{G^{(i)} : i = 1, \ldots, n\}$, ApproxLink predicts the scores of queried entries in the tensor.

- *Transductive learning over product graph* (**TOP**) [27]: TOP is designed for one-class classification. Given a binary incomplete initial tensor $\mathcal{Y}^0 \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ with ones and zeros representing the observed positive entries and missing entries respectively, and n knowledge graphs $\{G^{(i)} : i = 1, \ldots, n\}$, TOP detects the queried positive entries in the tensor.

- *Canonical polyadic decomposition* (**CP**): Tensor $\mathcal{Y}^0 \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ which is either noisily complete or incomplete with zeros representing the missing entries, is decomposed into n component matrices as in Equation (2.2). The component matrices are then used to construct the queried entries in the tensor.

- *Graph regularized CP* (**GraphCP**) [22]: Given n knowledge graphs $\{G^{(i)} : i = 1, \ldots, n\}$, an initial tensor $\mathcal{Y}^0 \in \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ which is either noisily complete or incomplete with zeros representing the missing entries is decomposed into n component matrices by solving a *least square* problem with *cross-mode* regularization using TPG. The component matrices are then used to construct the queried entries in the tensor.

- *Fast cross-layer dependency inference on multi-layered networks* (**FASCINATE**) [10]: Given a multi-layered network which contains the within-layer connections in $\{G^{(i)} : i = 1, \ldots, n\}$ and incomplete cross-layer connections in $\{R_{i,j} \in \mathbb{R}_+^{I_i \times I_j} : \forall i, j \in [1, n] \text{ and } i < j\}$, FASCINATE predicts the scores of the missing entries in $R_{i,j}$ using graph-regularized matrix factorization. Since FASCINATE does not learn a tensor or its decomposition directly, we treat the non-negative component matrices output by FASCINATE as the CPD-components to construct the n-way tensor for comparisons.

- *Spectral methods for multiple PPI network alignment* (**IsoRankN**) [12]: Given n PPI networks $\{G^{(i)} : i = 1, \ldots, n\}$ of n species, and BLAST sequence similarities $\{R_{i,j} \in \mathbb{R}_+^{I_i \times I_j} : \forall i, j \in [1, n] \text{ and } i < j\}$ between proteins from each pair of species, IsoRankN finds a global alignment of the n PPI networks based on spectral clustering on the induced graph of bipartite alignment scores.

- *Backbone extraction and merge strategy for multiple PPI network alignment* (**BEAMS**) [71]: Given n PPI networks $\{G^{(i)} : i = 1, \ldots, n\}$ of n species, and BLAST sequence similarities $\{R_{i,j} \in \mathbb{R}_+^{I_i \times I_j} : \forall i, j \in [1, n] \text{ and } i < j\}$ between proteins from each pair of species, BEMAS finds a global alignment of the n PPI networks by solving a combinatorial optimization problem with a heuristic approach.

Implementation details

The graph regularization hyperparameter α defined in section 2.4.2 was chosen from the pool {0.001, 0.1, 0.9, 0.99} for LowrankTLP, ApproxLink and TOP; rank $\left\lceil \sqrt[n]{k} \right\rceil$ approximation was applied to each individual graph for ApproxLink and TOP to guarantee the approximated TPG has the same or larger rank than k.

For better scalability, we adopted the first-order method ADAM [72] based on the *all-at-once* optimization [73, 74] to minimize the objective functions of CP and GraphCP. The component matrices were randomly initialized; the stopping criteria was chosen to be $||\nabla f(\boldsymbol{x}^t)||_2 \leq 10^{-3}||\nabla f(\boldsymbol{x}^0)||_2$, where $\boldsymbol{x}^t$ denotes the stack of all the vectorized component matrices in the t-th iteration; the maximum number of iterations was set to 1000. Note that, for GraphCP the gradient scale of the *cross-mode* regularization term increases much faster than the gradient scale of the decomposition term, as the tensor order (the number of graphs) increases. Therefore, when the tensor order is high, the graph hyperparameter α as defined in [22] tends to be set very small. Unless otherwise stated, we chose α from $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ as suggested in [22] and r (tensor rank) from $\{10, 50, 100\}$ for both CPD-based methods in Task 1; for Task 2 the CPD-rank r is equal to the rank of the initial CPD-form tensor and is chosen by PCA to cover at least 90% of the spectral energy in the stacking matrix R defined in Section 2.4.1.

We implemented FASICNATE[1] using its open-source package, with the graph hyperparameter α selected from $\{0.1, 0.5, 0.9, 1\}$ as suggested in [10].

The baselines IsoRankN[2] and BEAMS[3] were developed specifically for PPI network alignment. We downloaded and ran the original packages to obtain the alignment scores with the graph hyperparameter α selected from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ as suggested in their packages.

All the experiments were performed using our server with Intel(R) Xeon(R) CPU E5-2450 with 32 cores (2.10GHz) in 2 CPUs and 196GB of RAM. All the baseline methods except IsoRankN and BEAMS were implemented using MATLAB R2018b.

[1] `https://github.com/chenannie45/FASCINATE`
[2] `http://cb.csail.mit.edu/cb/mna/`
[3] `http://webprs.khas.edu.tr/~cesim/BEAMS.tar.gz`

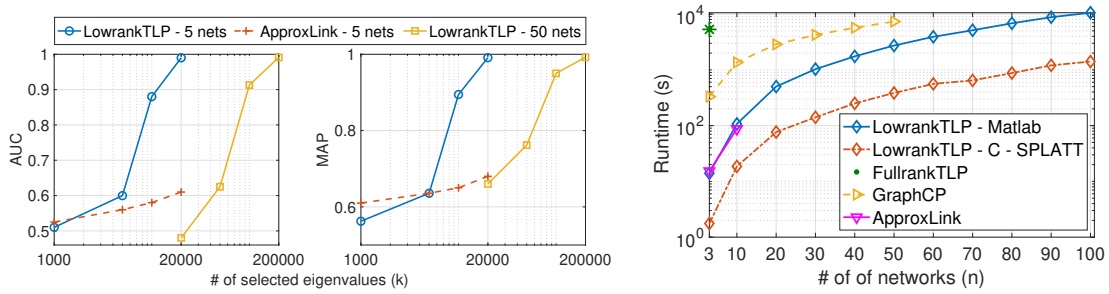


Figure 4.3: **Simulation results.**
(A) Effectiveness comparisons by varying TPG ranks. **(B)** Efficiency and scalability comparisons. The curves are truncated if the methods are not scalable.

4.4.2 Simulations

Synthetic graphs were generated to evaluate the performance and the scalability. We started with a graph of density 0.1 and size I to generate n distinct graphs by randomly permuting 10% of edges from the common "ancestor" graph so that they share similar structures that can be utilized for matching the multi-way associations. The inputs are the n graphs and a sparse n-way tensor $\mathcal{Y}^0 \in \mathbb{R}^{I \times \ldots \times I}$ with $I/2$ (half) of its diagonal entries set to 1s. We set the other $I/2$ diagonal entries and $I/2$ randomly sampled off-diagonal entries to 0.9 and treated them as positive and negative test samples, respectively. The outputs are scores of the I test entries after label propagation, which can be used to distinguish the positive and negative classes based on the assumption that the nodes indexed by the diagonal entries of the tensor should have high similarities since they come from the same "ancestor" graph and this information should be captured by the TPG.

- ***Effectiveness***: We compared LowrankTLP with ApproxLink, CP and GraphCP using the same sparse tensor $\mathcal{Y}^0$ as input. For fair comparisons, we fixed $\alpha = 0.1$ for LowrankTLP and used the best hyperparameters for all the baseline methods. The area under the curve (AUC) and mean average precision (MAP) are the evaluation metrics. Each experiment was repeated five times and the average

performances were reported. Table 4.3 shows that LowrankTLP clearly outperforms all the baselines to learn multi-way associations among 5 and 10 graphs. The prediction of the CPD-based methods is almost random, which is not surprising given the fact that tensor $\mathcal{Y}^0$ is extremely sparse; this observation also agrees with the previous observations that the accuracy of tensor decomposition degrades severely when only a small fraction of entries is observed [22, 75]. Note that ApproxLink performs better than the CPD-based methods, which implies that label propagation is a more robust approach for sparse inputs than tensor decomposition for hyperlink prediction. Figure 4.3(A) shows that the LowrankTLP outperforms ApproxLink in different TPG ranks, which validates the advantage of our optimization formulation (4.1). It also shows that LowrankTLP requires only a moderate rank $k \geq 10,000$ when $n = 5$, and achieves a high performance with $k \geq$ 100,000 when $n = 50$, whereas ApproxLink is not applicable to such a large number of graphs.

- ***Efficiency and scalability***: We further compared the runtime of the MATLAB implementation of LowrankTLP using Tensor Toolbox [56] version 2.6 and the parallel implementation using SPLATT library [62] (described in Section 4.3.3) with the baseline methods applicable to the knowledge graphs. We chose a small tensor rank $r = 10$ for GraphCP for time efficiency. The TPG rank $k = \frac{10^4}{5}n$ was chosen for LowrankTLP which achieves AUC ≈ 0.9 empirically. In Figure 4.3(B), we observe that the parallel LowrankTLP results in a speedup of about one order of magnitude compared to the MATLAB version. The parallel implementation of LowrankTLP improved the runtime to 10^3s compared with 10^4s by the sequential implementation to align 100 graphs of size 1000 each. ApproxLink has a similar runtime as sequential LowrankTLP on 3 and 10 graphs, while it is not applicable for more graphs due to the exponential growth of the number of components as discussed in Section 4.3.4. The empirical runtime of GraphCP is worse than LowrankTLP even if the theoretical time complexity for computing the gradient is fast as analyzed in Table 4.1.

Table 4.3: **Effectiveness comparison in simulations.**

		LowrankTLP	ApproxLink	GraphCP	CP
5 nets	AUC	0.990	0.610	0.540	0.549
	MAP	0.991	0.679	0.539	0.528
10 nets	AUC	0.942	0.554	0.536	0.520
	MAP	0.952	0.672	0.533	0.527

4.4.3 Predicting multi-way associations in scientific publications

We downloaded the DBLP dataset of scientific publication records from AMiner (Extraction and Mining of Academic Social Networks) [76]. We built three graphs: Author $\times$ Author graph $G^{(1)} = (V^{(1)}, E^{(1)})$, Paper $\times$ Paper graph $G^{(2)} = (V^{(2}, E^{(2)})$ and Venue $\times$ Venue graph $G^{(3)} = (V^{(3)}, E^{(3)})$. In $G^{(1)}$, the edge weight is the count of papers that both authors have co-authored; in $G^{(2)}$, the edge weight is the number of times both papers were cited by another paper; and in $G^{(3)}$, the edge weight is calculated using Jaccard similarity between the vectors of the two venues whose dimensions are a bag of citations. After filtering the nodes with zero and low degrees in each graph, we finally obtained $|V^{(1)}| = 13{,}823$ and $|E^{(1)}| = 266{,}222$; $|V^{(2)}| = 11{,}372$ and $|E^{(2)}| = 4{,}309{,}772$; $|V^{(3)}| = 10{,}167$ and $|E^{(3)}| = 46{,}557{,}116$, similar to the dataset used in [27]. Given the natural relationship that a paper is written by an author, and published in a specific venue, we built 1) the initial tensor $\mathcal{Y}^0$ with 12,066 positive multi-way associations in the form (Author, Paper, Venue) for all the tensor-based methods; and 2) a multi-layered network with three node types for FASCINATE as described in Section 4.4.1.

We first performed 5-fold cross-validation with 3-fold training triples, 1-fold validation triples to select the best hyperparameters and 1-fold test triples for all the methods, using the 12,066 positive triples together with the same number of randomly sampled negative triples. In each training fold of FASCINATE, the cross-layer connections correspond to the validation and test triples were held out. Figure 4.4(A) shows the performance comparisons with standard deviations on all the 5 test folds. We observed that LowrankTLP clearly outperforms the baselines in every fold, and the methods utilizing graph information perform consistently better than CP, which does not use graph information, demonstrating that associations among the tensor entries carried by the manifolds in the knowledge graphs are leveraged to enhance the prediction performance. We also randomly sampled 0.1%, 10%, 50% and 90% of positive triples as training data

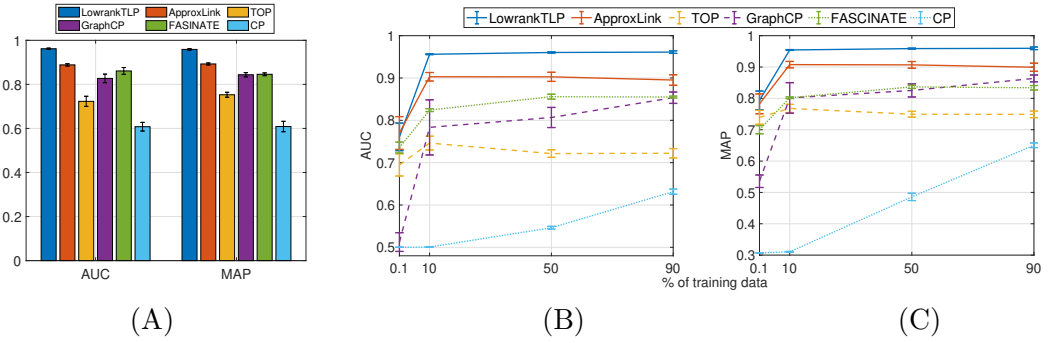


Figure 4.4: **DBLP results.**
(A) The performance of 5-fold cross-validation. The average and standard deviation across the 5 folds are shown. **(B)** & **(C)** The performance of using various percentages of training data. The average and standard deviation are shown for each percentage across different random samplings.

to test the rest of the positive triples together with the same number of randomly sampled negative triples. The random samplings are repeated 5 times for each percentage. Using the optimal hyperparameters chosen from the previous 5-fold cross-validation, we compared the performance of all the methods on various percentages of training/test data. Figure 4.4(B)&(C) show that LowrankTLP consistently outperforms all the baselines in every training percentage. Remarkably, both LowrankTLP and ApproxLink achieve average AUC ≈ 0.76 and MAP ≈ 0.8 when there are only 0.1% of training data; LowrankTLP, ApproxLink, TOP and FASCINATE are more robust to sparse input, comparing with GraphCP which is based on tensor decomposition.

The runtime of all the tensor-based multi-relational learning methods (LowrankTLP, ApproxLink, TOP and GraphCP) on DBLP dataset is also compared in Table 4.5. The observation is similar to the simulation where LowrankTLP and ApproxLink are more efficient than GraphCP and TOP. The efficiency of LowrankTLP and ApproxLink appears to be in the same scale while ApproxLink is more efficient since the DBLP tensor is only 3-way.

Table 4.4: **Performance of CT image alignment.**

	LowrankTLP					
	$k = 10$	$k = 10^2$	$k = 10^3$	$k = 10^4$	**CPD-form $\mathcal{Y}^0$**	**GraphCP**
4 images	0.59	0.91	0.91	0.91	0.61	0.73
5 images	0.67	0.80	0.89	0.89	0.66	0.75
6 images	0.78	0.78	0.84	0.84	0.69	0.78
7 images	0.75	0.73	0.80	0.83	0.72	0.76

4.4.4 Alignment of CT scan images

We obtained a dataset of 134 CT scan images of an anonymized female patient. The scans were acquired on a Philips Brilliance Big Bore CT Scanner, and each image has 512 $\times$ 512 pixels with a slice thickness of 3mm. We used a subset of 26 images which contain the same set of four segmented regions manually annotated by a radiologist. When working with CT scan images, the radiologist is interested in matching the segmented regions across the images. We represent this situation by aligning a set of sampled spots across the images to detect if they belong to the same type of segmented region. To construct a graph for each CT image, we first sampled from each segmented region in each image a number (proportional to the region size) of spots. Then, we calculated the similarity between the spots using the RBF function: $s(x_i, x_j) = \exp(-\frac{||x_i - x_j||^2}{\sigma})$ if $\phi(x_i) \neq \phi(x_j)$ and otherwise 1, where x_i and x_j are the coordinates of the two spots; $\phi(x)$ represents the region where the spot x is located; $\sigma = 10$ is the width of RBF function. The bipartite similarity scores between the spots in two different images were obtained by the color density difference between the spots, calculated using a RBF function with $\sigma = 10^{-2}$. The initial tensor $\mathcal{Y}^0$ was then generated in CPD-form using these cross-image spots similarity matrices. For example, to align 7 images, $\binom{7}{2} = 21$ similarity matrices were generated. In this setting, the number of spots can be different across the images. Therefore, it is possible that one spot in an image is matched to more than one spots in another image after the alignment.

The set of query tuples were selected if the color densities between each pair of the spots in a tuple are all above a threshold. The alignment accuracy was measured by the top-1 match of each spot. Specifically, for each spot in the first graph, we took the subtensor of dimension $(n-1)$ associated with the entry in the first dimension to find the entry of the highest score in the subtensor. Then, we checked if the features of the

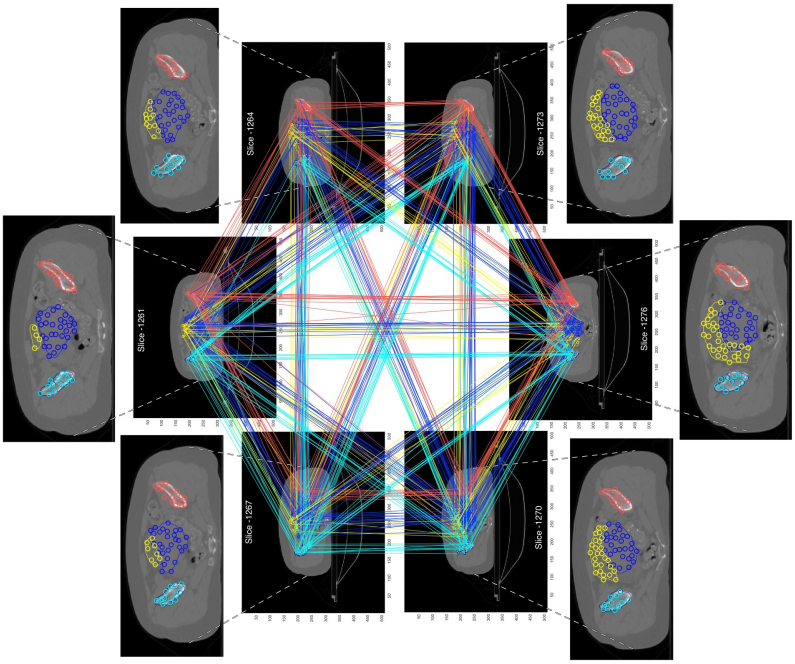
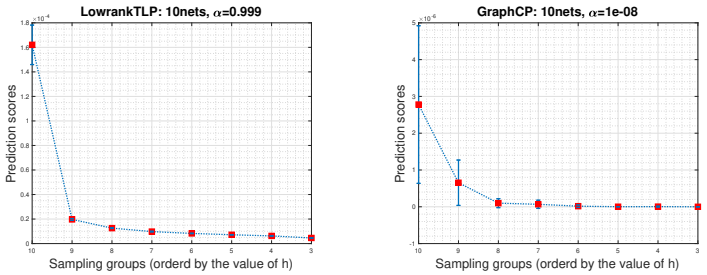


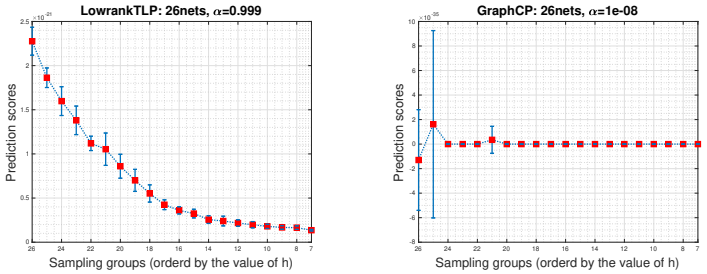
Figure 4.5: **Example of aligning 6 CT scan images.**
Each type of segmented region in the images is represented by a different color in the alignment. The links connect all the pairs of the spots in a 6-tuple with one from each image to represent one alignment.

aligned spots from all the other images in the maximum entry were the same as the spot in the first dimension. Table 4.4 shows the comparisons of LowrankTLP and GraphCP, using CPD-form tensor $\mathcal{Y}^0$ as their initialization. With $k \geq 100$, LowrankTLP achieves much higher accuracy than GraphCP in almost all the cases. It is also interesting that with $k = 10,000$, LowrankTLP is able to align 7 images with an accuracy of 0.83, which means 83% of the spots in the first graph is perfectly matched with a spot from the same type of segmented region in each of the other 6 images. An example of 6 aligned images is shown in Figure 4.5. It is clear that the aligned spots are consistent across the images.

More importantly, to further measure the scalability of LowrankTLP on a larger number of real graphs, we performed an additional evaluation by aligning 10 and 26 CT scan images. Since it is not computationally feasible to enumerate every entry of



(A) Alignment of 10 CT scans



(B) Alignment of 26 CT scans

Figure 4.6: **Results of aligning 10 and 26 CT scan images.**
The average and standard deviation of the prediction scores of the 10^4 n-tuples in each sampling group ordered by the homogeneity score h is shown.

the 10-way tensor and the 26-way tensor, we generated a list of candidate n-tuples for performance evaluation. Given n images to be aligned, we randomly sampled a list of n-tuples of spots. The n-tuples were then grouped by their homogeneity score h, where h is defined as the maximum number of spots that are from the same type of segmented region in the n-tuple. For example, if there are 3, 4, 10 and 9 spots in a 26-tuple from the 4 types of regions, respectively, the homogeneity score of this 26-tuple will be $\max(3, 4, 10, 9) = 10$. Based on the homogeneity score, we generated sampling groups of varying h to evaluate the alignment of $n = 10$ and $n = 26$ images. We expect that the sampling groups of larger h also receive higher prediction scores on the n-tuples in the groups. GraphCP is the only baseline that is both applicable and scalable in this experiment for comparison.

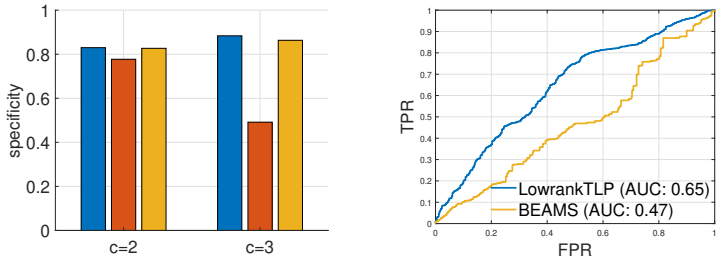
The average and standard deviation of the prediction scores for each sampling group are shown in Figure 4.6. In the alignment of 10 images shown in Figure 4.6(A), we observe that LowrankTLP generates a much larger average score for $h = 10$ compared

with the sampling groups with $h < 10$, and the average score decreases consistently and monotonically as h decreases. GraphCP is also able to identify the group of $h = 10$ but the variance is large and a flatter tail is observed after $h = 6$. In the alignment of 26 images shown in Figure 4.6(B), GraphCP completely fails to distinguish the most significant group $h = 26$ from the other groups, whereas LowrankTLP maintained the same clear decreasing trend as h decreases. This comparison implies LowrankTLP is more applicable to high-order TPG of a large number of graphs in real-world problems. As discussed in Section 4.4.1, the graph hyperparameter α of GraphCP was set to be very small when the number n of graphs is large.

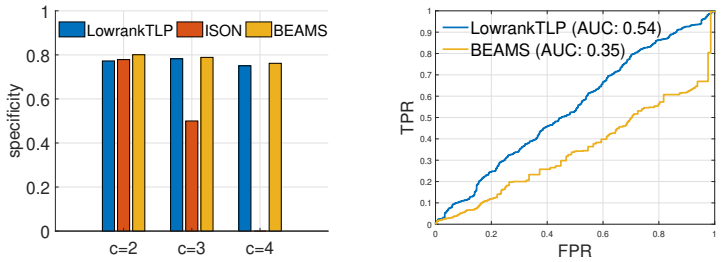
The runtime of all the tensor-based multi-relational learning methods (LowrankTLP, ApproxLink, TOP and GraphCP) on the CT scan dataset is also compared in Table 4.5. Note that ApproxLink and TOP are not applicable on this dataset and thus, no running time is reported in the comparison.

4.4.5 Alignment of PPI Networks

We downloaded the IsoBase dataset [12, 14, 77], containing protein-protein interactions (PPI) networks for five species: *H. sapiens* (HS), *D. melanogaster* (DM), *S. cerevisiae* (SC), *C. elegans* (CE) and *M. musculus* (MM). The *M. musculus* network only contains 776 interactions and is dropped from the analysis. After removing proteins with no association in the PPI networks, there are 10,403, 7,396, 5,524 and 2,995 proteins and 109,822, 49,991, 165,588 and 9,711 interactions in the HS, DM, SC and CE PPI networks, respectively. The dataset also contains cross-species protein sequence similarities as BLAST Bit-values for all the pairs of species. Similar to the CT scan experiment, we generated the input tensor $\mathcal{Y}^0$ in CPD-form whose dimensions are matched with the number of proteins in the corresponding species, by using the pairwise BLAST sequence similarity scores. In addition, the annotations of the proteins with 37,463 gene ontology (GO) terms below level five of GO are also provided for evaluation. We generated a set of query tuples of proteins with high sequence similarity between all the protein pairs in the tuple. These tuples can then be classified as true multi-way associations if all the annotated proteins in the tuple share at least one common GO term, and otherwise false multi-way associations. The experiments were performed using three species (HS, DM and SC) and four species by adding CE. Around 3M tuples were generated among



(A) Alignment of HS/DM/SC networks



(B) Alignment of HS/DM/SC/CE networks

Figure 4.7: **Results of PPI network alignment.**
In both (A) and (B), the figure on the left shows the specificity of the detected clusters containing different number of species, and the figure on the right shows the AUC curves between consistent and inconsistent query entries by prediction among the clusters reported by BEAMS.

three species and about 163M among four species.

Similar to the post-processing in the evaluation in [71], after applying LowrankTLP to generate the prediction scores for all the query tuples, the tuples were sorted for a greedy merge as protein clusters for standard evaluation of PPI network alignment. A cluster of size n is defined as a set of proteins with at least one protein from each of the n species. The greedy merge scans the tuples and adds the tuple that only contains proteins not seen yet as a new cluster. Otherwise, the proteins that are already in some other clusters are removed from the tuple, and the remaining proteins are added as a smaller cluster. In the evaluation, the specificity is defined as the ratio between the number of consistent clusters and the number of annotated clusters, where an annotated cluster is a cluster in which at least two proteins are associated to at least one GO term, and a consistent cluster is the one in which all of its annotated proteins share at least

one GO term. In the left plot in Figure 4.7(A), for both clusters of size 2 and 3, LowrankTLP performs better than both BEAMS and IsorankN in the alignment of the three networks. The left plot in Figure 4.7(B) shows that LowrankTLP performed similarly or slightly worse than BEAMS in every cluster size in the alignment of four networks. IsorankN is not able to detect any cluster of size 4.

To further compare LowrankTLP with BEAMS, we analyzed the detailed ranking of the annotated clusters with at least one protein from each species reported by BEAMS. Specifically, we enumerated all the tuples containing one protein from each species from each cluster and then applied LowrankTLP to calculate the scores of all the tuples in the output tensor. We re-ranked these tuples by the scores and annotated them as consistent or inconsistent multi-way associations by GO annotations. The AUC by their rankings is shown in the right plots in Figure 4.7. In both the three-network alignment and the four-network alignment, LowrankTLP ranks the consistent multi-way associations above the inconsistent multi-way associations with AUC larger than 0.5. Notice that since we only check the very top of the predictions (those predicted as true multi-way associations), the AUC is less than 0.5 for BEAMs results.

Table 4.5: **Runtime comparison using real datasets.**

	DBLP		CT scan	
	CPU time (s)	AUC	CPU time (s)	Accuracy
LowrankTLP	799	0.96	13	0.91
GraphCP	3440	0.82	64	0.73
ApproxLink	492	0.88	N/A	N/A
TOP	71500	0.72	N/A	N/A

4.5 Discussion

In this chapter, we introduced a new algorithm LowrankTLP to improve the scalability and performance of label propagation on tensor product graphs for multi-relational learning. The theoretical analysis in Chapter 6 will show that the globally optimal solution of the proposed optimization formulation minimizes an estimating error bound for recovering the true tensor from the noisy initial tensor for multiple graph alignment, and will provide the data-dependent transductive Rademacher bound for binary hyperlink prediction. In the experiments, we demonstrated that LowrankTLP well approximates

label propagation on the normalized tensor product graph to achieve both the better scalability and performance. We also demonstrated that LowrankTLP, capable of taking either a sparse tensor or a CPD-form tensor as input, is a flexible approach to meet the requirements of multi-relational learning problems in a wide range of applications. In all the experiments, we also observed that it does not require a huge rank to achieve a good prediction performance even if the size of a tensor product graph is exponential of the size of the individual graphs. This observation supports that the direct and efficient analysis of the entire spectrum of the tensor product graph is a better approach.

Chapter 5

Multi-relational Learning for the Imputation of Spatially-resolved Transcriptomes

5.1 Introduction

Dissection of complex genomic architectures of heterogeneous cells and how they are organized spatially in tissue are essential for understanding the molecular and cellular mechanisms underlying important phenotypes. For example, each tumor is a mixture of different types of proliferating cancerous cells with changing genetic materials [78]. The cancer cell sub-populations co-evolve in the micro-environment formed around their spatial locations. It is important to understand the cell-cell interactions and signaling as well as the functioning of each individual cell to develop effective cancer treatment and eradicate all cancer clones at their locations [79]. Conventional gene expression analyses have been limited to low-resolution bulk profiling that measures the average transcription levels in a population of cells. With single-cell RNA sequencing (scRNA-seq) [80–82], single cells are isolated with a capture method such as fluorescence-activated cell sorting (FACS), Fluidigm C1 or microdroplet microfluidics and then the RNAs are captured, reverse transcribed and amplified for sequencing the RNAs barcoded for the individual origin cells [83, 84]. While scRNA-seq is useful for detecting

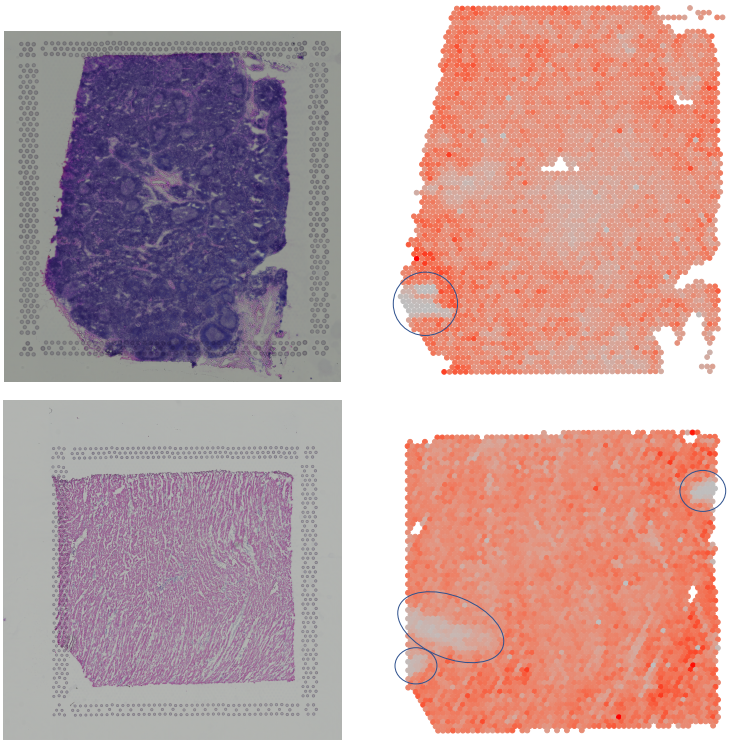
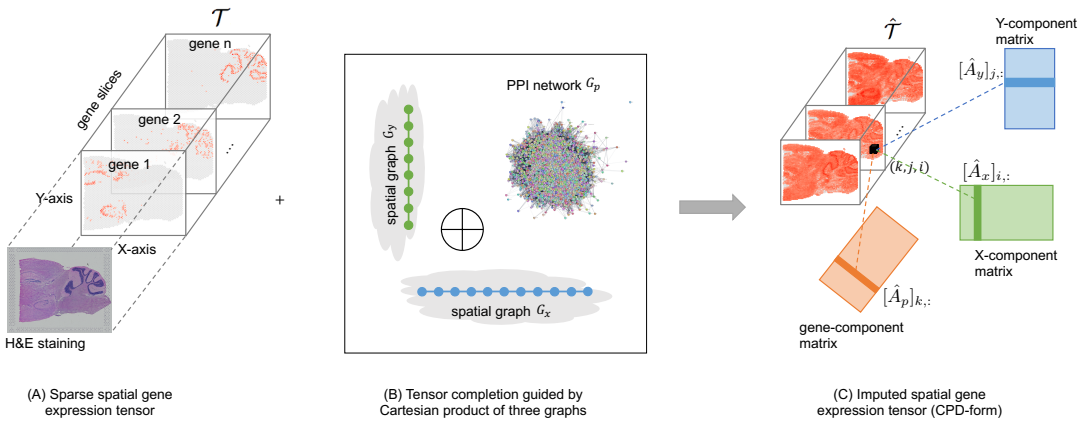


Figure 5.1: **Spatial regions with failed RNA fixing and permeabilization.** The H&E images are shown on the left, and the heatmaps of the total RNA count at each spot are shown on the right. The regions with irregularly low RNA count are annotated by the circles.

the cell heterogeneity in a tissue sample, it does not provide the spatial information of the isolated cells. To map cell localization, earlier in-situ hybridization methods such as FISH [85], FISSEQ [86], smFISH [87] and MERFISH [88] were developed to profile up to a thousand targeted genes in pre-constructed references with single-molecule RNA imaging. Based on in-situ capturing technologies, more recent spatial transcriptomics RNA sequencing (sptRNA-seq) [89–92] combines positional barcoded arrays and RNA sequencing with single-cell imaging to spatially resolve RNA expressions in each measured spot in the spatial array [89, 93–95]. These new technologies have transformed the transcriptome analysis into a new paradigm for connecting single-cell molecular profiling to tissue micro-environment and the dynamics of a tissue region [96–98].

With in-situ capturing technology, RNAs are captured and sequenced in the spots on the spatial genomic array aligned to the locations on the tissue. For example, spatial transcriptomics technology based on 10x Genomics Visium kit reports the number of copies of RNAs by counting unique molecular identifiers (UMIs) in the read-pairs mapped to each gene [99]. There are still significant technical difficulties. First, in-situ capturing has a low RNA capture efficiency. The earlier spatial transcriptomics technology's detection efficiency is as low as 6.9% and 10x Genomics Visium has only a slightly improved efficiency [100]. In addition, the sample preparation requires highly specific handling of tissue sections. The spots in some tissue regions might entirely fail to fix and permeabilize RNAs due to various possible issues in preparing tissue sections. A few examples of such regions are shown in Figure 5.1. Thus, sptRNA-seq data often only provides an incomplete profiling of the gene expressions over the spatial regions of the tissue. Similarly, in scRNA-seq data analysis, the missing gene expressions are called dropout events, which refer to the false quantification of a gene as unexpressed due to the failure in amplifying the transcripts during reverse-transcription [101]. It has been shown in previous studies on scRNA-seq data that normalizations will not address the dropout effects [20,99]. In the literature, many imputation methods such as Zero-inflated factor analysis (ZIFA) [19], Zero-Inflated Negative Binomial-based Wanted Variation Extraction (ZINB-WaVE) [21] and BISCUIT [20] have been developed to impute scRNA-seq. While these methods are also applicable to impute the spatial gene expressions, they ignore a unique characteristic of sptRNA-seq data, which is the spatial information among the gene expressions in the spatial array, and do not fully take advantage of the functional relations among genes for more reliable joint imputation.

To provide a more suitable method for imputation of spatially-resolved gene expressions, we introduce FIST: Fast Imputation of Spatially-Resolved Transcriptomes by Graph Regularized Tensor Completion. FIST is a tensor completion model regularized by a product graph as illustrated in Figure 5.2. FIST models sptRNA-seq data as a 3-way sparse tensor in genes (p-mode) and the (x, y) spatial coordinates (x-mode and y-mode) of the observed gene expressions (Figure 5.2(A)). As shown in Figure 5.2(B), a protein-protein interaction network models the interactions between pairs of genes in the gene mode, and the spatial graph is modeled by a product graph of two chain graphs for columns (x-mode) and rows (y-mode) in the grid to capture the spatial relations



(A) Sparse spatial gene expression tensor

(B) Tensor completion guided by Cartesian product of three graphs

(C) Imputed spatial gene expression tensor (CPD-form)

Figure 5.2: **Tensor-based imputation of the spatial transcriptomes.** **(A)** The input sptRNA-seq data is modeled by a 3-way sparse tensor in genes (p-mode) and the (x, y) spatial coordinates (x-mode and y-mode) of the observed gene expressions. H&E image is also shown to visualize the cell morphologies aligned to the spots. **(B)** A protein-protein interaction network and a spatial graph are integrated as a product graph for tensor completion. The spatial graph is also a product graph of two chain graphs for columns (x-mode) and rows (y-mode) in the grid.**(C)** After the imputation, the CPD-form of the complete tensor can be used to impute any missing gene expressions, e.g. the entry (k, j, i) can be reconstructed as the sum of the element-wise multiplications of the three components $[\hat{A}_p]_{k,:}$, $[\hat{A}_y]_{j,:}$ and $[\hat{A}_x]_{i,:}$.

among the (x, y) spots. The Cartesian product of these graphs with prior knowledge of gene functions and the spatial relations among the capture spots are then introduced as a regularization of tensor completion to obtain the canonical polyadic decomposition (CPD) of the tensor. The imputation of the unobserved entries can then be derived by reconstructing the entries in the completed tensor shown in Figure 5.2(C). In the experiments, we comprehensively evaluated FIST on ten 10x Genomics Visium spatial genomics datasets by comparison with widely used methods for single-cell RNA sequencing data imputation. We also analyzed a mouse kidney dataset with more functional interpretation of the gene clusters obtained by the imputed gene expressions to detect highly relevant functions in the clusters expressed in three kidney tissue regions, cortex, outer stripe of the outer medulla (OSOM) and inner stripe of the outer medulla (ISOM).

5.2 Methods

In this section, we first describe the task of spatial gene expression imputation, and next propose to solve the task of spatial gene expression imputation using the graph-regularized tensor completion model described in Section 2.3. We then present a fast iterative algorithm FIST to solve the optimization problem defined to optimize the model. We also provide the convergence analysis of proposed algorithm in Section 6.3. The notations which will be used for the derivations in the forthcoming sections are summarized in Table 5.1.

Table 5.1: **Notations of the sptRNA-seq data.**

Notation	Definition
G_x, G_y	Spatial chain graphs of (x, y) coordinates
G_p	Protein-protein interaction (PPI) network
n_x, n_y, n_p	Number of nodes in G_x, G_y, G_p
$W_x \in \mathbb{R}_{[0,1]}^{n_x \times n_x}, W_y \in \mathbb{R}_{[0,1]}^{n_y \times n_y}, W_p \in \mathbb{R}_{[0,1]}^{n_p \times n_p}$	Adjacency matrix of G_x, G_y, G_p
$L_x \in \mathbb{R}^{n_x \times n_x}, L_y \in \mathbb{R}^{n_y \times n_y}, L_p \in \mathbb{R}^{n_p \times n_p}$	Graph Laplacian of G_x, G_y, G_p
$\mathfrak{G}(x, y, p)$	Cartesian product of G_x, G_y, G_p
$\mathfrak{W}(x, y, p) \in \mathbb{R}_{[0,1]}^{n_x n_y n_p \times n_x n_y n_p}$	Adjacency matrix of $\mathfrak{G}(x, y, p)$
$\mathfrak{L}(x, y, p) \in \mathbb{R}^{n_x n_y n_p \times n_x n_y n_p}$	Graph Laplacian of $\mathfrak{G}(x, y, p)$
$\mathcal{T} \in \mathbb{R}_{+}^{n_p \times n_y \times n_x}$	Incomplete spatial gene expression tensor
$\hat{\mathcal{T}} \in \mathbb{R}_{+}^{n_p \times n_y \times n_x}$	Complete spatial gene expression tensor
$\mathcal{M} \in \mathbb{R}_{[0,1]}^{n_p \times n_y \times n_x}$	Binary mask tensor
$\hat{A}_x \in \mathbb{R}_{+}^{n_x \times r}$, $\hat{A}_y \in \mathbb{R}_{+}^{n_y \times r}$, $\hat{A}_p \in \mathbb{R}_{+}^{n_p \times r}$	CPD component matrices of $\hat{\mathcal{T}}$
$\mathbf{vec}(\mathcal{T}) \in \mathbb{R}^{n_x n_y n_p \times 1}$	Rearrange $\mathcal{T}$ to be a vector

5.2.1 Imputation of spatial gene expressions by tensor modeling

Let $\mathcal{T} \in \mathbb{R}_{+}^{n_p \times n_y \times n_x}$ be the 3-way sparse tensor of the observed spatial gene expression data as show in Figure 5.2(A), with the missing gene expressions represented as zeros, where n_p denotes the total number of genes, n_x and n_y denote the dimensions of the x and y spatial coordinates of the spatial transcriptomics array. Our goal is to learn a complete spatial gene expression tensor $\hat{\mathcal{T}} \in \mathbb{R}_{+}^{n_p \times n_y \times n_x}$ from $\mathcal{T}$ as illustrated in Figure 5.2(C). The advantage of the tensor representation is to incorporate the 2-D spatial x-mode and y-mode such that the grid structure is preserved within the columns and the rows of the spatial array in the tensor, which contains useful spatial information.

We propose to model the task of spatial gene expression imputation as a tensor completion problem, which can be solved by the graph-regularized tensor decomposition method describe in Section 2.3. The key ideas of modeling the task of spatial gene expression imputation are i) the inferred complete spatial gene expression tensor $\hat{\mathcal{T}}$ is regularized to integrate the spatial arrangements of the spots in the tissue array and the functional relations among the genes; ii) the observed part in $\mathcal{T}$ is also required to be preserved in $\hat{\mathcal{T}}$ as the completion task requires; and iii) the inferred tensor $\mathcal{T}$ is compressed as the CPD-form $\hat{\mathcal{T}} = [\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!]$ for space and time efficiencies. The novel optimization formulation is shown below in Proposition 5.2.1.

Proposition 5.2.1. *The complete spatial gene expression tensor $\hat{\mathcal{T}} \in \mathbb{R}^{n_p \times n_y \times n_x}$ can be obtained by solving the following optimization problem:*

$$\begin{aligned} \underset{\{\hat{A}_p, \hat{A}_y, \hat{A}_x\}}{\textit{minimize}} \quad & \frac{1}{2}||\mathcal{M} \circledast (\mathcal{T} - \hat{\mathcal{T}})||_{\mathcal{F}}^2 + \frac{\lambda}{2}\mathbf{vec}(\hat{\mathcal{T}})^T \mathfrak{L}(x, y, p)\mathbf{vec}(\hat{\mathcal{T}}) \\ \textit{subject to} \quad & \hat{\mathcal{T}} = [\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!] \\ & \hat{A}_p \geq 0, \hat{A}_x \geq 0, \hat{A}_y \geq 0. \end{aligned} \tag{5.1}$$

where $\lambda \in [0, 1]$ is a model hyperparameter, and $||.||_{\mathcal{F}}$ denotes the Frobenius norm of a tensor.

The optimization problem in Equation (5.1) is in the same form as the Equation (2.1), except that the component matrices $\hat{A}_p$, $\hat{A}_y$, and $\hat{A}_x$ are required to be non-negative to guarantee the imputed gene expression tensor $\hat{\mathcal{T}}$ is non-negative. We use the Laplacian $\mathfrak{L}(x, y, p)$ of the Cartesian product graph (described in Section 2.2) to regularize the tensor $\hat{\mathcal{T}}$.

- **Cartesian product graph regularization**

  Two useful assumptions to introduce prior knowledge for inferring the tensor are *1) the spatially adjacent spots should share similar gene expressions, and 2) the expressions of two genes are likely highly correlated if they share similar gene functions [102, 103].* We introduce a spatial graph and a protein-protein interaction (PPI) network into the model.

  We first encode the spatial information in two undirected unweighted chain graphs $G_x = (V_x, E_x)$ and $G_y = (V_y, E_y)$. There are $|V_x| = n_x$ nodes in G_x where n_x is

the number of the spatial coordinates along the x-axis of the spatial array. Two nodes in G_x are connected by an edge if they are adjacent along the x-axis. The connections in G_y can be similarly defined to encode the y-coordinates of the tissue.

We also incorporate the topological information of a PPI network download from BioGRID 3.5 [30] to use the functional modules in the PPI network. We denote the PPI network as $G_p = (V_p, E_p)$ which contains $|E_p|$ experimentally documented physical interactions among the $|V_p| = n_p$ proteins. We then use the Cartesian product $\mathfrak{G}(x, y, p) = (V, E)$ of the three individual graphs G_x, G_y and G_p to regularize the elements in $\hat{\mathcal{T}}$, where $|V| = n_x n_y n_p$.

By introducing the term $\mathbf{vec}(\hat{\mathcal{T}})^T \mathfrak{L}(x, y, p)\mathbf{vec}(\hat{\mathcal{T}})$ in Equation (5.1), the inferred gene expression values in $\hat{\mathcal{T}}$ are ensured to be smooth over the manifolds of the product graph $\mathfrak{G}(x, y, p)$, such that a pair of tensor entries $\hat{\mathcal{T}}_{a_p,a_y,a_x}$ and $\hat{\mathcal{T}}_{b_p,b_y,b_x}$ share similar values if the (a_x, a_y, a_p)-th and (b_x, b_y, b_p)-th nodes are connected in $\mathfrak{G}(x, y, p)$. A connection implies that the x-coordinate a_x and b_x is adjacent or y-coordinate a_y and b_y is adjacent or gene a_p and gene b_p are connected in the PPI, with the two other dimensions fixed. Using Cartesian product graph is a more conservative strategy to connect multi-way associations in a high-order graph as we have shown in Chapter 3, since only replacing one of the dimensions by the immediate neighbors is allowed to create connections. Note that it also possible to use tensor product graph or strong product graph, but there could be too many connections to provide meaningful connectivity in the product graph for helpful regularization.

It is known that genes' connectivities in PPI network correlate with their co-expressions. We justified this hypothesis on the spatial transcriptomics data by examining the relation between the connectivity in PPI network and the co-expression in spatial locations among the genes in the 10 different 10x Genomics Visium spatial genomics datasets used in this study. The results of this analysis are shown in Figure 5.3. We observed higher co-expressions between the genes that are connected with less hops in the PPI, which supports our assumptions.

5.2.2 FIST Algorithm

In this section, we propose an efficient iterative algorithm Fast Imputation of Spatially-Resolved Transcriptomes by Graph Regularized Tensor Completion (FIST) to find its local optimal solution using the multiplicative updating rule [47], based on derivatives of $\hat{A}_p$, $\hat{A}_y$ and $\hat{A}_x$. Without loss of generality, we only show the derivations with respect to $\hat{A}_p$, and provide the FIST algorithm in Algorithm 5.1.

We first bring the equality constraint $\hat{\mathcal{T}} = [\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!]$ in Equation (5.1) into the objective function, and rewrite the objective function as

$$\begin{aligned} \mathcal{J} &= \mathcal{J}_1 + \lambda \mathcal{J}_2 \\ \mathcal{J}_1 &= \frac{1}{2}||\mathcal{M} \circledast (\mathcal{T} - [\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!])||_{\mathcal{F}}^2 \\ \mathcal{J}_2 &= \frac{1}{2}\mathbf{vec}([\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!])^T \mathfrak{L}(x, y, p)\mathbf{vec}([\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!]) \end{aligned} \tag{5.2}$$

The partial derivative of $\mathcal{J}_1$ with respect to $\hat{A}_p$ can be computed as

$$\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p} = (\mathcal{M}_{(1)} \circledast \hat{\mathcal{T}}_{(1)} - \mathcal{M}_{(1)} \circledast \mathcal{T}_{(1)})(\hat{A}_x \odot \hat{A}_y). \tag{5.3}$$

Note that the term $\mathcal{M}_{(1)} \circledast \hat{\mathcal{T}}_{(1)}$ in Equation (5.3) implies we only need to compute the entries in $\hat{\mathcal{T}}$ which correspond to the non-zero entries (indices of the observed gene expression) in $\mathcal{M}$. The rest of the computation in Equation (5.3) involves the well-known MTTKRP (matricized tensor times Khatri-Rao product) [70] operation, which is in the form of $\mathcal{X}_{(1)}(\hat{A}_x \odot \hat{A}_y)$, and can be computed in $O(r|\mathcal{X}|)$ if $\mathcal{X}$ is a sparse tensor with $|\mathcal{X}|$ nonzeros, and $\hat{A}_x$ and $\hat{A}_y$ have r columns. Thus, the overall time complexity of computing Equation (5.3) is $O(r|\mathcal{M}|)$.

Following the derivations in Section 3.2.2, we obtain the partial derivatives of the second term $\mathcal{J}_2$ as

$$\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p} = \hat{A}_p(\Phi_x \circledast \Theta_y + \Phi_y \circledast \Theta_x) + L_p\hat{A}_p(\Phi_x \circledast \Phi_y), \tag{5.4}$$

where $\Phi_i = \hat{A}_i^T\hat{A}_i$, and $\Theta_i = \hat{A}_i^T L_i \hat{A}_i$, for all $i \in \{x, y, p\}$. It is not hard to show that the complexity of computing the Equation (5.4) is $O(\sum_{i\in\{x,y,p\}}(r^2 n_i + r n_i^2))$.

Next, we combine $\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}$ and $\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}$ to obtain the overall derivative as

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \hat{A}_p} &= \frac{\partial \mathcal{J}_1}{\partial \hat{A}_p} + \lambda(\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}) \\ &= [\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^+ - [\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^- + \lambda([\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^+ - [\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^-),\end{aligned} \tag{5.5}$$

where $[\frac{\mathcal{J}_i}{\partial \hat{A}_p}]^+$ and $[\frac{\mathcal{J}_i}{\partial \hat{A}_p}]^-$ are non-negative components in $\frac{\mathcal{J}_i}{\partial \hat{A}_p}$, which are defined below as

$$[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^+ = (\mathcal{M}_{(1)} \circledast \hat{\mathcal{T}}_{(1)})(\hat{A}_x \odot \hat{A}_y), \tag{5.6}$$

$$[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^- = (\mathcal{M}_{(1)} \circledast \mathcal{T}_{(1)})(\hat{A}_x \odot \hat{A}_y), \tag{5.7}$$

$$[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^+ = \hat{A}_p(\Phi_x \circledast \Theta_y^D + \Phi_y \circledast \Theta_x^D) + D_p\hat{A}_p(\Phi_x \circledast \Phi_y), \tag{5.8}$$

$$[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^- = \hat{A}_p(\Phi_x \circledast \Theta_y^W + \Phi_y \circledast \Theta_x^W) + W_p\hat{A}_p(\Phi_x \circledast \Phi_y), \tag{5.9}$$

where $\Theta_i^D = \hat{A}_i^T D_i \hat{A}_i$ and $\Theta_i^W = \hat{A}_i^T W_i \hat{A}_i$, for all $i \in \{x, y, p\}$. According to Equation (5.5), the objective function $\mathcal{J}$ objective will monotonically decrease under the following multiplicative updating rule,

$$[\hat{A}_p]_{a,b} \leftarrow [\hat{A}_p]_{a,b}\left(\frac{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^-_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^-_{a,b}}{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^+_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^+_{a,b}}\right), \tag{5.10}$$

where $[\hat{A}_p]_{a,b}$ denotes the (a, b)-th element in matrix $\hat{A}_p$. Similarly, we can derive the update rule for $[\hat{A}_x]_{a,b}$ and $[\hat{A}_p]_{a,b}$ as follows,

$$[\hat{A}_y]_{a,b} \leftarrow [\hat{A}_y]_{a,b}\left(\frac{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_y}]^-_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_y}]^-_{a,b}}{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_y}]^+_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_y}]^+_{a,b}}\right), \tag{5.11}$$

$$[\hat{A}_x]_{a,b} \leftarrow [\hat{A}_x]_{a,b}\left(\frac{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_x}]^-_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_x}]^-_{a,b}}{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_x}]^+_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_x}]^+_{a,b}}\right). \tag{5.12}$$

We then propose an efficient iterative algorithm FIST in Algorithm 5.1 to find the local optimum of the proposed graph regularized tensor completion problem with time

complexity $O(r|\mathcal{M}| + \sum_{i\in\{x,y,p\}}(r^2 n_i + r n_i^2))$. FIST takes the incomplete spatial gene expression tensor $\mathcal{T}$, PPI network and spatial chain graphs as input, and outputs the inferred CPD representation of the complete spatial gene expression tensor $\hat{\mathcal{T}}$, via solving the optimization problem defined in Proposition 5.2.1 with the multiplicative updating rule (line 4-6 in Algorithm 5.1) based on the tensor calculus in Equations (5.6)-(5.9). With the efficient tensor computation in Equations (5.6)-(5.9), the algorithm can avoid computing the full Cartesian product graph and tensors, and break down the calculus into the computation on the individual graphs and the sparse tensors. Therefore, FIST is proven to be a scalable method, which only requires the space $O(|\mathcal{T}| + |\mathcal{M}|)$ to store the sparse tensors, $O(\sum_{i\in\{x,y,p\}} |E_i|)$ to store the graphs, and $O(\sum_{i\in\{x,y,p\}} r n_i)$ to store the component matrices. Thus, the overall space complexity is $O(|\mathcal{T}| + |\mathcal{M}| + \sum_{i\in\{x,y,p\}}(|E_i| + r n_i))$. The theoretical convergence analysis of FIST is given in Section 6.3.

Algorithm 5.1: FIST: Fast Imputation of Spatially-Resolved Transcriptomes by Graph Regularized Tensor Completion

Data: 1) spatial gene expression tensor $\mathcal{T} \in \mathbb{R}_{+}^{n_p \times n_y \times n_x}$, 2) binary mask tensor $\mathcal{M} \in \mathbb{R}_{[0,1]}^{n_p \times n_y \times n_x}$ which indicates the observed part in $\mathcal{T}$, 3) protein-protein interaction (PPI) network G_p and 4) hyper parameter λ.

Result: The low-rank matrices $\hat{A}_p, \hat{A}_y$ and $\hat{A}_x$, which form the CPD representation of the inferred spatial gene expression tensor $\hat{\mathcal{T}} = [\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!] \in \mathbb{R}_{+}^{n_p \times n_y \times n_x}$.

1 Construct the spatial chain graphs G_x and G_y as described in the text.

2 Randomly initialize $\hat{A}_p \in \mathbb{R}_{+}^{n_p \times r}$, $\hat{A}_y \in \mathbb{R}_{+}^{n_y \times r}$ and $\hat{A}_x \in \mathbb{R}_{+}^{n_x \times r}$ as non-negative matrices.

3 **while** *not converge* **do**

4 update $\hat{A}_p$ by Equation (5.10).

5 update $\hat{A}_y$ by Equation (5.11).

6 update $\hat{A}_x$ by Equation (5.12).

7 **end**

5.3 Experiments

In the experiments, we first summarize the baseline methods for performance comparisons in section 5.3.1, and describe the data preparation and performance measures in Section 5.3.2 and 5.3.3. Then, we evaluate the accuracy of FIST and baseline methods for the imputation of sptRNA-seq data with three different metrics in Section 5.3.4, and study the role of Cartesian product graph in Section 5.3.5. Next, in Section 5.3.6, we analyze the imputed spatial gene expressions in the Mouse Kidney Section dataset to show several interesting gene clusters revealing functional characteristics of the three tissue regions, corex, OSOM and ISOM. Finally, in Section 5.3.7. we apply FIST to an additional dataset to demonstrate its broad applicability.

5.3.1 Baseline methods and implementations

To benchmark the performance of FIST, we compared it with three matrix factorization (MF)-based methods (with graph regularizations) and a nearest neighbors (NN)-based method, which have been applied to impute various types of biological data including the imputation of dropouts in single-cell RNA sequencing (scRNA-seq) data. Note that NMF-based methods have been shown to be effective for learning latent features and clustering of high-dimension sparse genomic data [104].

- **ZIFA**: Zero-inflated factor analysis (ZIFA) [19] factorizes the single cell expression data $Y \in \mathbb{R}^{N\times D}$ where N and D denote the number of single cells and genes respectively, into a factor loading matrix $A \in \mathbb{R}^{K\times D}$ and a matrix $Z \in \mathbb{R}^{N\times K}$ which spans the latent low-dimensional space where dropouts can happen with a probability specified by an exponential decay associated with the expression levels. The imputed matrix can be computed as $\hat{Y} = ZA + \mu$, where $\mu \in \mathbb{R}^{1\times D}$ is the latent mean vector.

- **REMAP**: Since ZIFA is a probabilistic MF model which does not utilize the spatial and gene networks, we therefore also compare with REMAP [32], which was developed to impute the missing chemical-protein associations for the identification of the genome-wide off-targets of chemical compounds. REMAP factorizes the incomplete chemical-protein interactions matrix into the chemical and protein

low-rank matrices, which are regularized by the compound similarity graph and protein sequence similarity (NCBI BLAST [105]) graph respectively.

- **GWNMF**: Both ZIFA and REMAP are only applicable to the spot-by-gene matrix which is a flatten of a input tensor $\mathcal{T}$. Such flattening process assumes the spots are independent from each other, and thus loses the spatial information. To keep the spatial arrangements, we also apply MF to each $n_x \times n_y$ slice in tensor $\mathcal{T}$. Specially, we adopt the graph regularized weighted NMF (GWNMF) [39] method to impute each $n_x \times n_y$ gene slice. We let GWNMF use the same x-axis and y-axis graphs G_x and G_y as described in the previous section to regularize the MF.

- **Spatial-NN**: It has been observed that in sparse high-dimensional scRNA-seq data, constructing a nearest neighbor (NN) graph among cells can produce more robust clusters in the presence of dropouts because of taking into account the surrounding neighbor cells [106]. Such rationale has be considered in the clustering methods such as Seurat [107] and shared nearest neighbors (SNN)-Cliq [106], and can also be adopted to impute the spatial gene expression data. We introduce a SNN-based baseline Spatial-NN using neighbor averaging to compare with FIST. Specifically, to impute the missing expression of a target spot, Spatial-NN first searches its spatially nearest spots with observed gene expressions, then assign their average gene expression to the target spot.

We used the provided Python package[1] to experiment with ZIFA, and the provided MATLAB package[2] to experiment with REMAP. To apply both methods, we rearranged the data tensor $\mathcal{T} \in \mathbb{R}^{n_p \times n_y \times n_x}$ to a matrix $T \in \mathbb{R}^{N \times n_p}$, where $N = n_x n_y$ denotes the total number of spots. The spatial graph of REMAP is constructed by connecting two spots if they are spatially adjacent. REMAP adopts the same PPI network as the gene graph G_p as used by FIST. We used MATLAB to implement GWNMF and Spatial-NN ourselves to impute each gene slice $T_i \in \mathbb{R}^{n_x \times n_y}$ in $\mathcal{T}$. In the comparisons, the graph hyperparameter λ of FIST is selected from $\{0, 0.01, 0.1, 1\}$. The graph hyperparameters of REMAP and GWNMF are set by searching the grids from $\{0.1, 0.5, 0.9, 1\}$ and $\{0, 0.1, 1, 10, 100\}$ respectively as suggested in the original studies. Note

[1] `https://github.com/epierson9/ZIFA`
[2] `https://github.com/hansaimlim/REMAP`

Table 5.2: **Spatial transcriptome datasets from 10x Genomics.**

Dataset	Tissue section	Tensor dimensions	Density
HBA1	Human Breast Cancer (Block A Section 1)	$13,426 \times 60 \times 77$	0.093
HBA2	Human Breast Cancer (Block A Section 2)	$13,470 \times 58 \times 75$	0.100
HH	Human Heart	$7,487 \times 63 \times 70$	0.049
HLN	Human Lymph Node	$12,368 \times 61 \times 78$	0.088
MKC	Mouse Kidney Section (Coronal)	$12,264 \times 41 \times 56$	0.103
MBC	Mouse Brain Section (Coronal)	$13,570 \times 49 \times 74$	0.110
MB1P	Mouse Brain Serial Section 1 (Sagittal-Posterior)	$15,404 \times 62 \times 67$	0.115
MB2P	Mouse Brain Serial Section 2 (Sagittal-Posterior)	$12,497 \times 63 \times 65$	0.077
MB1A	Mouse Brain Serial Section 1 (Sagittal-Anterior)	$12,658 \times 59 \times 66$	0.105
MB2A	Mouse Brain Serial Section 2 (Sagittal-Anterior)	$12,295 \times 63 \times 66$	0.082

that different methods use different scales of graph hyperparameters since the gradients of their variables with respective to the regularization terms are in different scales. The optimal hyper-parameters are selected by the validation set for each method. For FIST, REMAP and GWNMF, we applied PCA on matrix $T \in \mathbb{R}^{N \times n_p}$ to determine the rank $r \in [200, 300]$ of the low-rank component matrices, such that at least 60% of the variance is accounted for by the top-r PCA components of T. The latent dimension K of ZIFA is set to 10 since it is time consuming to run ZIFA with a larger K. We also observed that increasing K from 10 to 50 does not show clear improvement on the imputation accuracy.

5.3.2 Preparation of spatial gene expression datasets

We downloaded the spatial transcriptomic datasets from 10x Genomics[3] , which is a collection of spatial gene expressions in 10 different tissue sections from mouse brain, mouse kidney, human breast cancer, human heart and human lymph node as listed in Table 5.2. All the sptRNA-seq datasets were collected with 10x Genomics Visium Spatial protocol (v1 chemistry) [91] to profile each tissue section with a high density hexagonal array with 4,992 spots to achieve a resolution of 55 µm (1-10 cells per spot). To fit a tensor model on the spatial gene expression datasets, we organized each of the 10 datasets into a 3-way tensor $\mathcal{T} \in \mathbb{R}^{n_p \times n_y \times n_x}$, where the (i, j, k)-th entry in $\mathcal{T}$ is the UMI count of the i-th gene at the (k, j)-th coordinate in the array. Note that the spots

[3] `https://support.10xgenomics.com/spatial-gene-expression/datasets/`

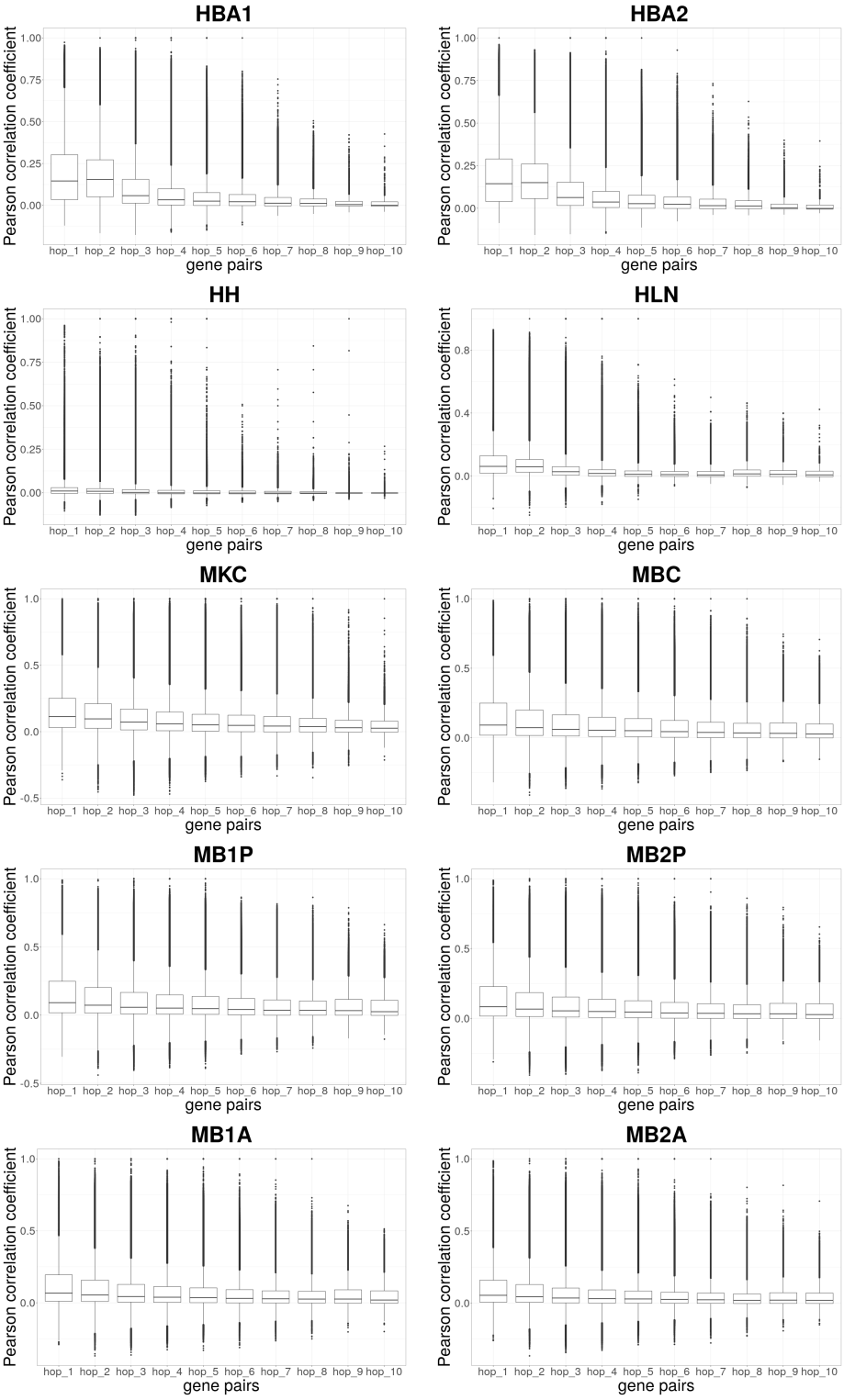


Figure 5.3: **PPI co-expression analysis.**
The Pearson correlation coefficients between expression values of k-hop gene pairs from PPI network are shown as box plots.

are arranged in a perfect grid in earlier spatial transcriptomic arrays and the rows and columns in the grid can be used directly as the coordinates (n_x,n_y). In the Visium array slide, the spots are arranged in a hivegrid. To map the spatial coordinates (n_x,n_y), we shifted the odd-numbered rows by a half of a spot for a convenient arrangement of the spots in the tensor without loss of generality. We set the entries in $\mathcal{T}$ to zeros if their UMI counts is lower than 3. We then removed the genes with no UMI counts across the spots, and removed the empty spots in the boundaries of the four sides in the H&E staining from $\mathcal{T}$. The log-transformation is finally applied to every entry of $\mathcal{T}$ as $\mathcal{T}_{i,j,k} \leftarrow \log(1+\mathcal{T}_{i,j,k})$. The sizes and densities of the 10 different spatial gene expression tensors after prepossessing are summarized in Table 5.2. Finally, we downloaded the full Homo sapiens and Mus musculus protein-protein interactions (PPI) networks from BioGRID 3.5 [30] as the gene network G_p to match with the genes in each dataset.

5.3.3 Evaluations and performance measures

We applied 5-fold cross-validation to evaluate the performance of imputing spatial gene expressions by spatial spots or genes as follows:

- **Spot-wise evaluation**: We chose 4-fold of the non-empty spatial spots for training and validation, and held out the rest 1-fold non-empty spatial spots as test data. When evaluating the expressions $\mathcal{T}_{:,j,k} \in \mathbb{R}^{n_p \times 1}$ in the (k,j)-th spatial spot, we set the vectors $\mathcal{T}_{:,j,k}$ and $\mathcal{M}_{:,j,k}$ in the input tensor $\mathcal{T}$ and mask tensor $\mathcal{M}$ to zeros to indicate that the expressions in this spot are unobserved; next, we use the learned low-rank matrices $\hat{A}_p, \hat{A}_y$ and $\hat{A}_x$ to construct the predicted gene expressions $\hat{\mathcal{T}}_{:,j,k}$.

- **Gene-wise evaluation**: For each gene, we chose 4-fold of its observed expressions (nonzeros in $\mathcal{T}$) for training and validation, and held out the rest 1-fold observed expressions as test data. When evaluating the 1-fold expressions in the i-th gene $\mathcal{T}_{i,:,:} \in \mathbb{R}^{n_y \times n_x}$, we set the corresponding entries in $\mathcal{T}_{i,:,:}$ and $\mathcal{M}_{i,:,:}$ in the input tensor $\mathcal{T}$ and mask tensor $\mathcal{M}$ to zeros, to indicate the expressions in this fold are unobserved; next, we use the learned low-rank matrices $\hat{A}_p, \hat{A}_y$ and $\hat{A}_x$ to construct the predicted gene expressions $\hat{\mathcal{T}}_{i,:,:}$.

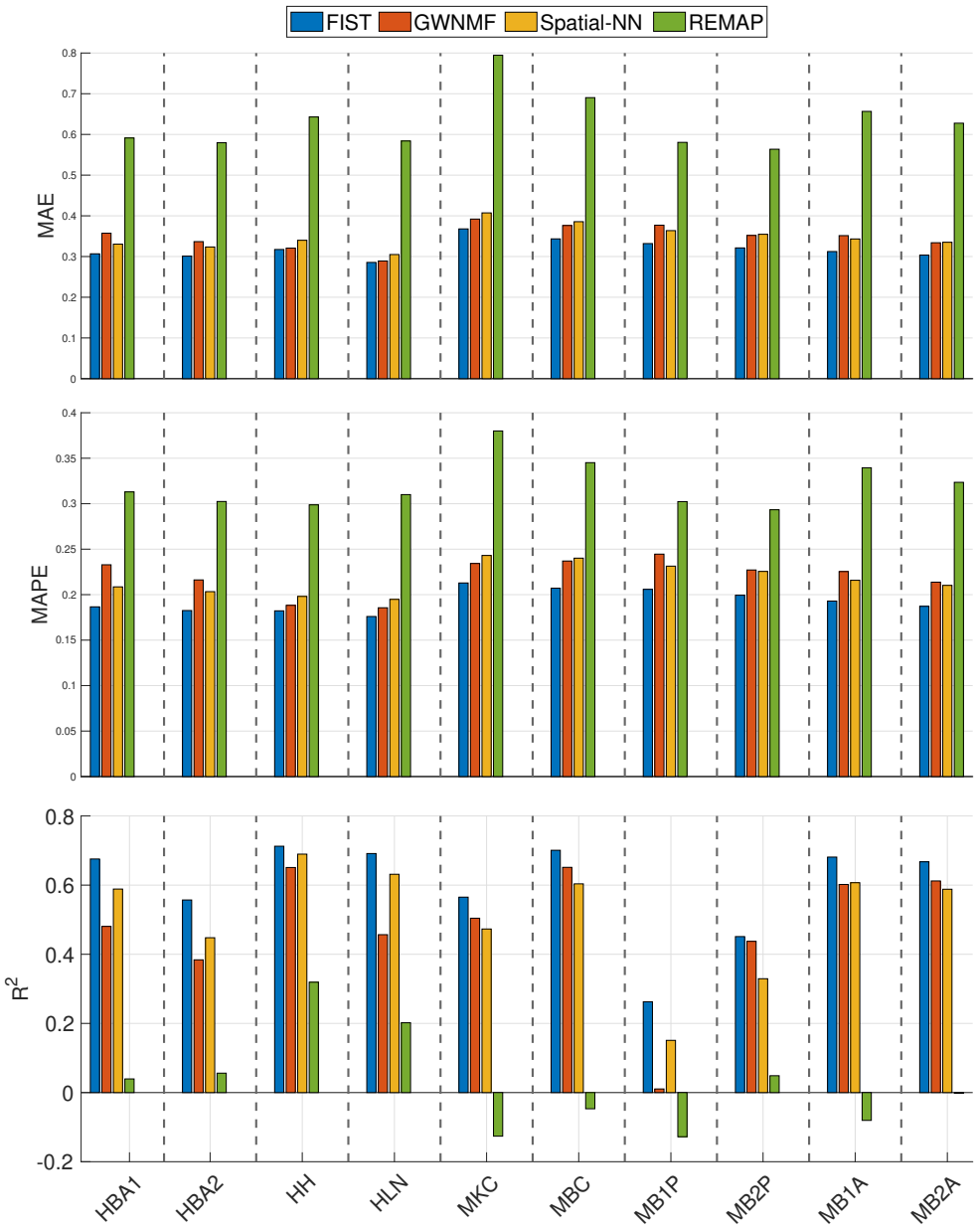


Figure 5.4: **Spot-wise cross-validation on 10x Genomics data.** The performances of the four compared methods on the 10 tissue sections are measured by 5-fold cross-validation. Each bar shows the mean of the imputation performance of one method on all the spatial spots. The result on each of the 10 datasets is shown in one vertical column separated by dashed lines.

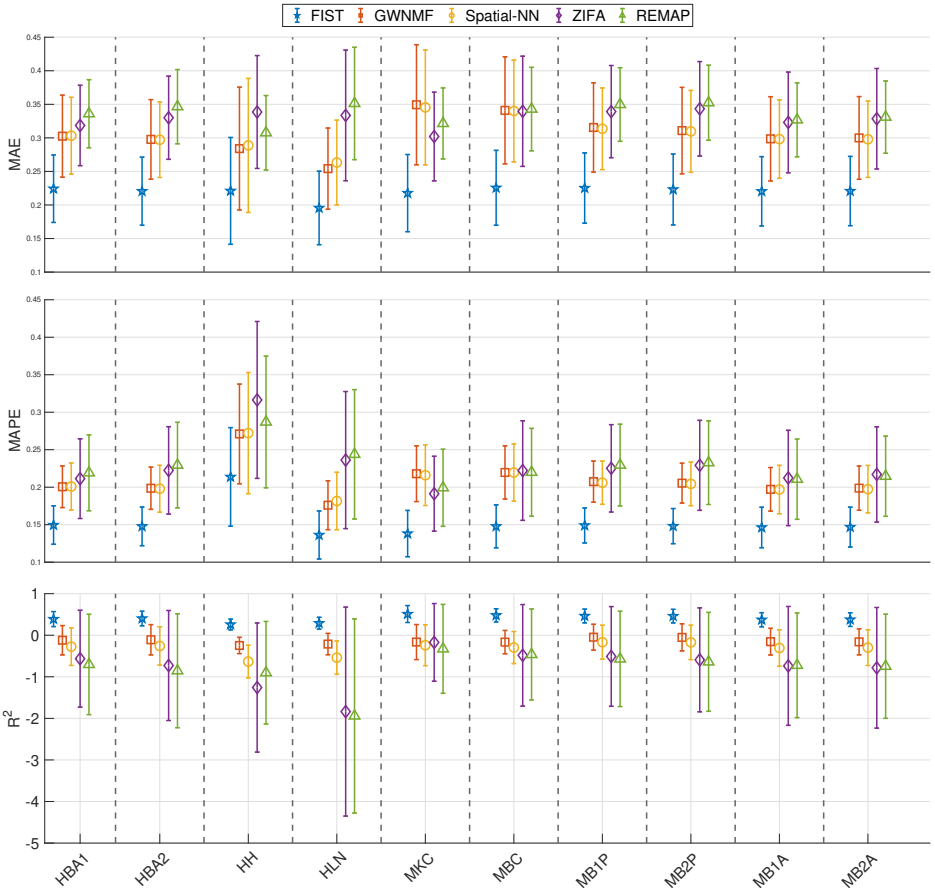


Figure 5.5: **Gene-wise cross-validation on 10x Genomics data.** The performances of the five compared methods on the 10 tissue sections are measured by 5-fold cross-validation. Each error bar shows the mean and variance of the imputation performance for one method on all the genes. The result on each of the 10 datasets is shown in one vertical column separated by dashed lines.

The hyper-parameter λ is optimized by the validation set for FIST and baseline methods. Denoting vectors $\mathbf{t} \in \mathbb{R}^{n\times 1}$ and $\hat{\mathbf{t}} \in \mathbb{R}^{n\times 1}$ as the true and predicted expressions in the held-out spatial spot $\mathcal{T}_{:,j,k}$ or the held-out entries in gene $\mathcal{T}_{i,:,:}$, the imputation performance is evaluated by the following three metrics,

- MAE (mean absolute error) $= \frac{1}{n}(\sum_{i=1}^{n} |\mathbf{t}_i - \hat{\mathbf{t}}_i|)$,
- MAPE (mean absolute percentage error) $= \frac{1}{n}(\sum_{i=1}^{n} |\frac{\mathbf{t}_i - \hat{\mathbf{t}}_i}{\mathbf{t}_i}|)$,
- R^2 (coefficient of determination) $= 1 - (\sum_{i=1}^{n}(\mathbf{t}_i - \hat{\mathbf{t}}_i)^2)(\sum_{i=1}^{n}(\mathbf{t}_i - \frac{\sum_{j=1}^{n}\mathbf{t}_j}{n})^2)^{-1}$.

We expect a method to achieve smaller MAE and MAPE and larger R^2 for better performance.

5.3.4 FIST significantly improves the accuracy of imputation

The performances of FIST and the baseline methods except for ZIFA in the spot-wise evaluation are compared in Figure 5.4. ZIFA was excluded from this spot-wise evaluation as it does not allow empty rows (spots) in the implementation of its package, and thus is not applicable to the prediction of the held-out test spots. The average performances of all the spatial spots using each of the 10 sptRNA-seq datasets are shown as bar plots. FIST consistently outperforms all the baselines with lower MAE and MAPE, and larger R^2 in all the 10 datasets. The performances of FIST and the baseline methods in the gene-wise evaluation are compared in Figure 5.5. The average and standard deviation of the prediction performances across all the genes are shown as error bar plots in Figure 5.5. Similar to the spot-wise evaluation, FIST clearly outperforms all the baselines with more robust performances across all the genes, as the variances in all the three evaluation metrics are also lower than the other compared methods. The observations suggest that FIST indeed performs better than the other methods in the imputation accuracy informed by the spatial information in the tensor model. It is also noteworthy that GWNMF, the MF method regularized by the spatial graph applied to each individual gene slice in tensor $\mathcal{T}$, outperforms the other baselines in almost all the datasets. This observation further confirms that the spatial patterns maintained in each gene slice is informative for the imputation task. It is clear that FIST outperforms

GWNMF with better use of the spatial information coupled with the functional modules of the PPI network G_p and the joint imputation of all the genes in the tensor $\mathcal{T}$.

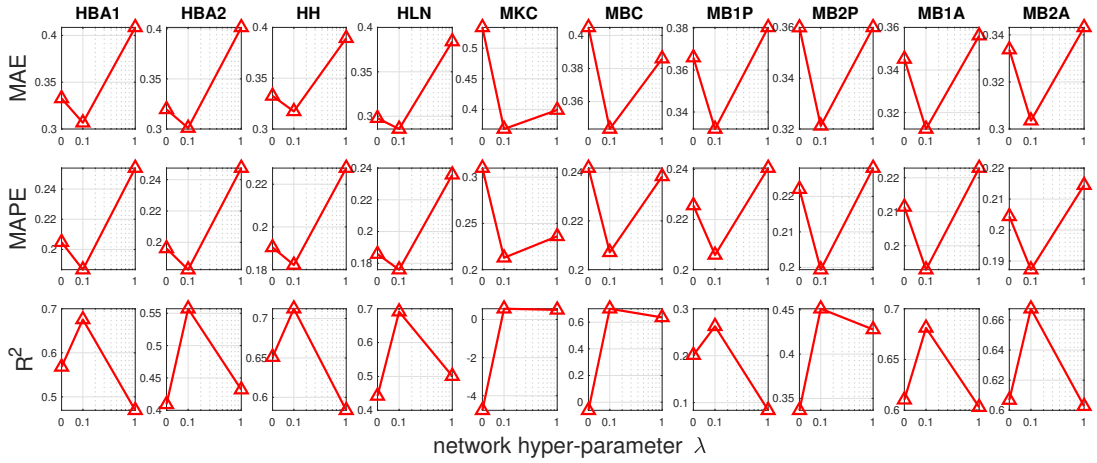
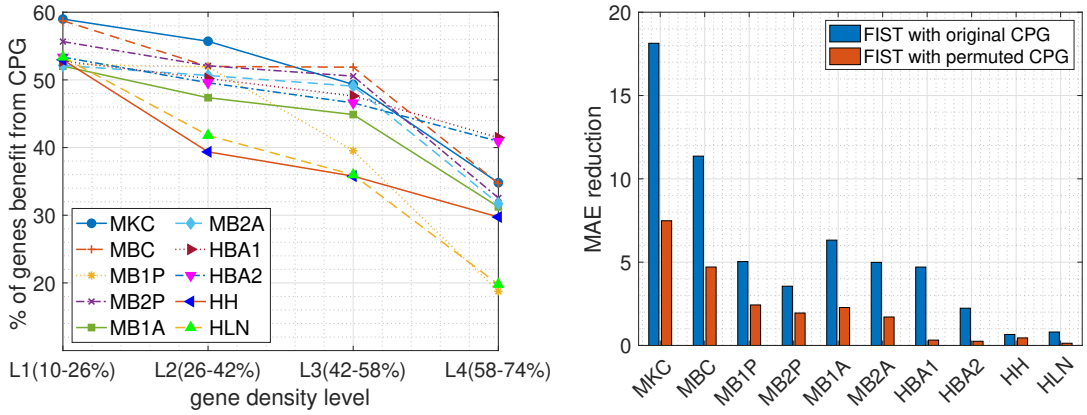


Figure 5.6: **Analysis of CPG regularization in spot-wise evaluation.** The plots show the imputation performance of FIST on the ten 10x Genomics datasets with varying network hyper-parameters in $\{0, 0.1, 1\}$ by MAE, MAPE and R^2.

5.3.5 Cartesian product graph regularization plays a significant role

To demonstrate that the Cartesian product graph regularization in FIST significantly improves the imputation accuracy, we showed in Figure 5.6 the performance of FIST in each of the 10 datasets by varying the graph hyper-parameter λ in the spot-wise evaluation. By increasing λ from 0 to 0.1 to put more belief on the graph information, we observe an appreciable reduction on the MAE and MAPE, and increase on R^2 across all the 10 datasets. The observation strongly suggests that the predictions by FIST are improved by leveraging the information carried in the CPG topology, and the belief on the graph information can be effectively optimized by using a validation set in the cross-validation strategy.

To further understand the associations between the CPG regularization and characteristics of the expressions of the genes, we analyzed the genes that are benefiting most from the regularization by the CPG in the gene-wise evaluation. In particular, in the grid search of the optimal λ weight on the CPG regularization term by the validation set, we count the percentage of the genes with optimal $\lambda = 0.01$ rather than 0,



(A) Imputation performance by density (B) Graph permutation test

Figure 5.7: **Analysis of CPG regularization in gene-wise evaluation.** (A) The percentages of genes benefit from the CPG are plotted by their densities in four different ranges. Each colored line represents one of the 10 datasets. (B) The total reduction of MAE using the original and permuted graphs are compared across the 10 tissue sections.

which means completely ignore the regularization. To correlate the improved imputations with the sparsity of the gene expressions, we divided all the genes into 4 equally partitioned groups (L1-4) ordered by their densities in the sptRNA-seq data, where L1 and L4 contain the sparsest and the densest gene slices, respectively. For each of the four density levels, we count the percentage of gene slices that benefit from the CPG regularization and plot the results in Figure 5.7(A). In the plots, there is a clear trend that the sparser a gene slice, the more likely it benefits from the CPG regularization in all the 10 datasets. In the densest L4 group, as low as 20% of the genes can benefit from the CPG regularization versus more than 50% in the sparest L1 group. This is understandable that there is less training information available for sparsely expressed genes (with more dropouts) and the spatial and functional information in the CPG can play a more important role in the imputation by seeking information from the gene's spatial neighbors or the functional neighbors in the PPI network. This observation is also consistent with the fact that the performance of tensor completion tends to degrade severely when only a very small fraction of entries are observed [22, 108], and therefore

those sparser gene slices tend to benefit more from the side information carried in the CPG.

We also compared the performance of FIST using the CPG of G_x, G_y and G_p with the one using a randomly permuted graph from the CPG. To generate the random CPG, we first generated three random graphs by permute G_x, G_y and G_p individually which also preserves the degree distributions of the original graphs, by randomly swapping the edges in each graph while keeping the degree of each node. Then we measured the performances of FIST using the original CPG and the CPG obtained from the permuted graphs by MAE reduction, which is the total reduction of MAE on all the genes by varying hyperparameter λ from 0 to 0.01 meaning not using the graph versus using the graph. The comparisons across all the 10 datasets are shown in Figure 5.7(B). We observe that the FIST using the original graphs receives much higher MAE reduction than the FIST using the permuted graphs. This observation suggests the topology in the original CPG carries rich information that is helpful for the imputation task beyond just the degree distributions preserved in the random graphs.

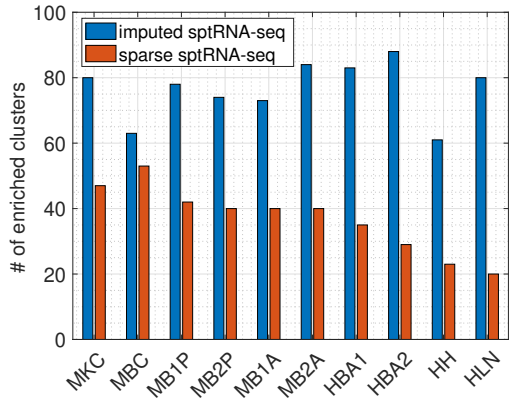


Figure 5.8: **Enrichment on the sparse and imputed sptRNA-seq data.** The total number of significantly enriched clusters (with at least one enriched GO term with FDR adjusted p-value < 0.05) in the 10 tissue sections are shown.

5.3.6 FIST recovers functionally relevant spatial patterns

To demonstrate that imputations by FIST can reveal spatial gene expression patterns with highly relevant functional characteristics among the genes in the spatial region, we performed comparative GO enrichment analysis of gene clusters detected with the imputed gene expressions. We conducted a case study on the Mouse Kidney Section data to further analyze the associations between the spatial gene clusters and the relevance between their functional characteristics and three kidney tissue regions, cortex, outer stripe of the outer medulla (OSOM) and inner stripe of the outer medulla (ISOM).

To validate the hypothesis that the imputed sptRNA-seq tensor $\widetilde{\mathcal{T}}$ given below

$$\widetilde{\mathcal{T}} = (1 - \mathcal{M}) \circledast \hat{\mathcal{T}} + \mathcal{T}$$

can better capture gene functional modules than the sparse sptRNA-seq tensor $\mathcal{T}$ does, we first rearranged both sptRNA-seq tensors into matrices $\widetilde{T} \in \mathbb{R}^{N \times n_p}$ and $T \in \mathbb{R}^{N \times n_p}$, where $N = n_x n_y$ denotes the total number of spots. We then applied K-means on each matrix to partition the genes into 100 clusters. Next, we used the enrichGO function in the R package clusterProfiler [109] to perform the GO enrichment analysis of the gene clusters. The total number of significantly enriched gene clusters (FDR adjusted p-value < 0.05) in each of the 10 tissue sections are shown in Figure 5.8, which clearly tells that K-means on the imputed sptRNA-seq data produces much more significantly enriched clusters across all the 10 tissue sections than the sparse sptRNA-seq data without imputation.

Finally, we conducted a case study on the Mouse Kidney Section and present the highly relevant functional characteristics in different tissues in mouse kidney detected with the imputations by FIST. For each of the 100 gene clusters generated by K-means as described above, we collapsed the corresponding gene slices in $\widetilde{\mathcal{T}}$ into a $n_x \times n_y$ matrix by averaging the slices to visualize the center of the gene cluster. We focus on 3 kinds of representative clusters in Figure 5.9 which match well with three distinct mouse kidney tissue regions: cortex, ISOM (inner stripe of outer medulla and OSOM (outer stripe of outer medulla). By investigating the enriched GO terms by the clusters (p-values shown in Table 5.3), we found their functional relevance to cortex, ISOM and OSOM regions. We found that the spatial gene cluster 9 which is highly expressed in cortex specifically enriched biological processes for the regulation of blood pressure (GO:0008217,

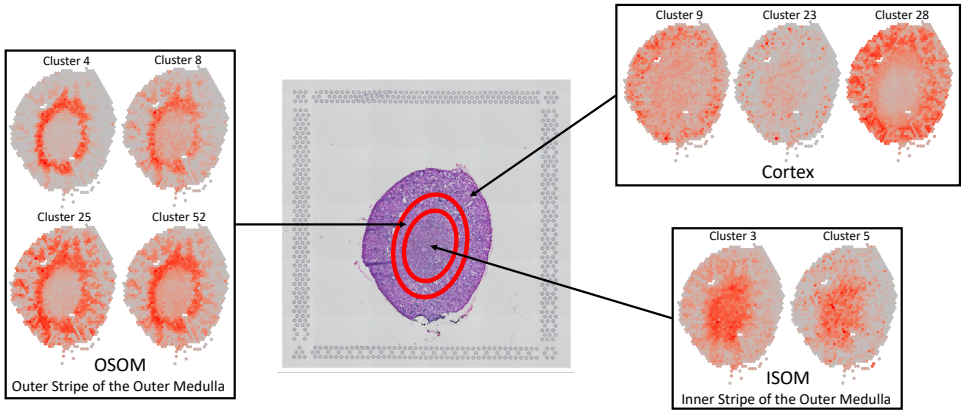


Figure 5.9: **FIST recovers spatial patterns on Mouse Kidney Section.** The H&E image of the mouse kidnesy section is shown in the middle with circles roughly separating the tissue area of Cortex, the outer stripe of the outer medulla (OSOM) and the inner stripe of the outer medulla (ISOM) from outer to inner regions. The gene expression patterns of the clusters in each of the three regions are grouped in the same box labeled by the region.

GO:0003073, GO:0008015 and GO:0045777) and transport/homeostasis of inorganic molecules (GO:0055067 and GO:0015672). The spatial gene cluster 23 and 28 which are also highly expressed in cortex enriched cellular pathways that are critical for the polarity of cellular membranes (GO:0086011, GO:0034763, GO:1901017, GO:0032413 and GO:1901380) and the transport of cellular metabolites (GO:1901605, GO:0006520, GO:0006790 and GO:0043648), respectively. These observations are consistent with previous studies reporting the regulation of kidney function by above listed biological processes in cortex [110–113]. In contrast, the pattern analysis of spatial gene expression in cluster 4, 8, 25 and 52 which are highly expressed in OSOM in kidney showed that catabolic processes of organic and inorganic molecules are specifically enriched such as GO:0015711, GO:0046942, GO:0015849, GO:0015718, GO:0010498, GO:0043161, GO:0044282, GO:0016054, GO:0046395, GO:0006631, GO:0072329, GO:0009062 and GO:0044242. These cellular processes are known to be active in renal proximal tubule which exists across cortex and OSOM [114–119]. Distinctively, the spatial gene clusters highly expressed in ISOM enriched pathways for nucleotide metabolisms (GO:0009150, GO:0009259 and GO:0006163) in cluster 3 and renal filtration (GO:0097205 and GO:0003094)

in cluster 5. Collectively, these observations demonstrate that FIST could identify physiologically relevant distinctive spatial gene expression patterns in the mouse kidney dataset. Further, it suggests that FIST can provide a high-resolution anatomical analysis of organ functions in sptRNA-seq data.

Table 5.3: **Functional terms enriched by spatial gene clusters.**

Region	Cluster	Significantly Enriched GO terms
Cortex	**Cluster 9**	GO:0003073 - regulation of systemic arterial blood pressure ($p = 9.1 \times 10^{-6}$) GO:0008217 - regulation of blood pressure ($p = 1.0 \times 10^{-4}$) GO:0055067 - monovalent inorganic cation homeostasis ($p = 4.3 \times 10^{-4}$) GO:0008015 - blood circulation ($p = 5.3 \times 10^{-3}$) GO:0045777 - positive regulation of blood pressure ($p = 5.8 \times 10^{-3}$) GO:0015672 - monovalent inorganic cation transport ($p = 2.3 \times 10^{-2}$)
	Cluster 23	GO:0086011 - membrane repolarization during action potential ($p = 2.2 \times 10^{-3}$) GO:0034763 - negative regulation of transmembrane transport ($p = 2.2 \times 10^{-3}$) GO:1901017 - negative regulation of potassium ion transmembrane transporter activity ($p = 2.4 \times 10^{-3}$) GO:0032413 - negative regulation of ion transmembrane transporter activity ($p = 2.7 \times 10^{-3}$) GO:1901380 - negative regulation of potassium ion transmembrane transport ($p = 3.4 \times 10^{-3}$)
	Cluster 28	GO:1901605 - alpha-amino acid metabolic process ($p = 4.8 \times 10^{-10}$) GO:0006520 - cellular amino acid metabolic process ($p = 6.4 \times 10^{-9}$) GO:0006790 - sulfur compound metabolic process ($p = 3.1 \times 10^{-6}$) GO:0043648 - dicarboxylic acid metabolic process ($p = 8.4 \times 10^{-6}$)
OSOM	**Cluster 4**	GO:0015711 - organic anion transport ($p = 7.7 \times 10^{-7}$) GO:0046942 - carboxylic acid transport ($p = 1.1 \times 10^{-4}$) GO:0015849 - organic acid transport ($p = 1.1 \times 10^{-4}$) GO:0015718 - monocarboxylic acid transport ($p = 5.0 \times 10^{-3}$)
	Cluster 8	GO:0010498 - proteasomal protein catabolic process ($p = 1.3 \times 10^{-3}$) GO:0006497 - protein lipidation ($p = 1.3 \times 10^{-3}$) GO:0042158 - lipoprotein biosynthetic process ($p = 1.3 \times 10^{-3}$) GO:0043161 - proteasome-mediated ubiquitin-dependent protein catabolic process ($p = 1.3 \times 10^{-3}$)
	Cluster 25	GO:0044282 - small molecule catabolic process ($p = 5.5 \times 10^{-19}$) GO:0016054 - organic acid catabolic process ($p = 1.0 \times 10^{-18}$) GO:0046395 - carboxylic acid catabolic process ($p = 1.0 \times 10^{-18}$) GO:0006631 - fatty acid metabolic process ($p = 2.9 \times 10^{-16}$) GO:0072329 - monocarboxylic acid catabolic process ($p = 9.6 \times 10^{-14}$) GO:0009062 - fatty acid catabolic process ($p = 1.0 \times 10^{-13}$) GO:0044242 - cellular lipid catabolic process ($p = 4.7 \times 10^{-11}$)
	Cluster 52	GO:0006732 - coenzyme metabolic process ($p = 1.2 \times 10^{-10}$) GO:0006520 - cellular amino acid metabolic process ($p = 1.6 \times 10^{-10}$) GO:1901605 - alpha-amino acid metabolic process ($p = 2.3 \times 10^{-9}$) GO:0044282 - small molecule catabolic process ($p = 2.1 \times 10^{-8}$) GO:0000096 - sulfur amino acid metabolic process ($p = 2.3 \times 10^{-7}$)
ISOM	**Cluster 3**	GO:0009150 - purine ribonucleotide metabolic process ($p = 7.4 \times 10^{-5}$) GO:0009259 - ribonucleotide metabolic process ($p = 7.4 \times 10^{-5}$) GO:0006163 - purine nucleotide metabolic process ($p = 7.4 \times 10^{-5}$) GO:0019693 - ribose phosphate metabolic process ($p = 7.4 \times 10^{-5}$) GO:0072521 - purine-containing compound metabolic process ($p = 7.4 \times 10^{-5}$)
	Cluster 5	GO:0048872 - omeostasis of number of cells ($p = 4.5 \times 10^{-5}$) GO:0030218 - erythrocyte differentiation ($p = 3.2 \times 10^{-3}$) GO:0034101 - erythrocyte homeostasis ($p = 3.2 \times 10^{-3}$) GO:0003094 - glomerular filtration ($p = 3.2 \times 10^{-3}$) GO:0097205 - renal filtration ($p = 3.2 \times 10^{-3}$)

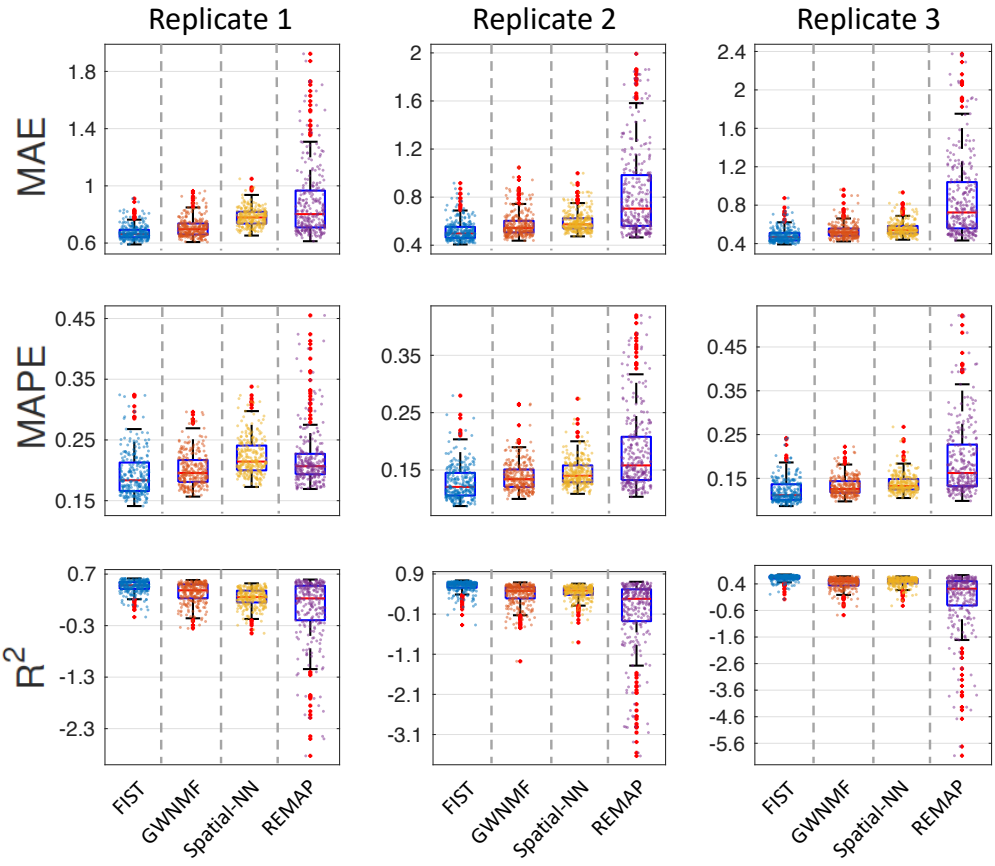


Figure 5.10: **Spot-wise imputation performance on mouse tissue replicates.** The performances of the four compared methods on the 3 replicates are measured by 5-fold cross-validation. The performance on each spatial spot is denoted by one dot in the box plots. The performances of different methods are shown in different colors.

5.3.7 Experiments on additional low-resolution datasets

To demonstrate that FIST is broadly applicable to impute the spatial gene expression data generated with different platforms, we performed additional experiments on spatially transcriptomics datasets from 3 replicates of mouse tissue (olfactory bulb) provided from an earlier study [120]. Developed before 10x Genomics Visium Spatial protocol, the spatial transcriptomics technology [120] applies an aligned array to profile tissue with both lower spot density and larger spot size (1,007 spots in total, and 200 µm between spots). The design achieves a resolution of 100 µm (10-40 cells per

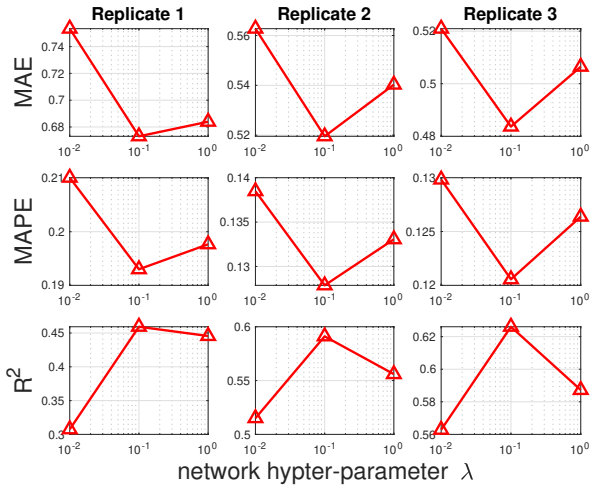


Figure 5.11: **Analysis of CPG regularization in the low-resolution data.**

spot). Similar to the experiments on the 10x Genomics data, we organized each of the 3 tissue replicates into a tensor $\mathcal{T} \in \mathbb{R}^{n_p \times n_y \times n_x}$, where $n_x = 33$ and $n_y = 35$ in all the 3 replicates, and n_p is 14,198, 13,818 and 138,40, respectively in replicate 1,2 and 3. The (i, j, k)-th entry in $\mathcal{T}$ is the RPKM value of the i-th gene at the (k, j)-th coordinate in the array.

We performed the spot-wise 5-fold cross-validation as we did in the 10x Genomics data to compare the performances of FIST and the same baseline methods. The distributions of MAE, MAPE and R^2 on all the spatial spots in each of the 3 tissue replicates are shown in Figure 5.10. Consistent with the observations in the previous Figure 5.4 and Figure 5.5, FIST clearly outperforms all the baselines with lower MAE and MAPE, and larger R^2 in all the 3 replicates. The results suggest that FIST has a potential to be applied to various spatial transcriptomics datasets of different resolution and sparseness to achieves better imputation performance by modeling the spatial data as tensors, and including the prior knowledge with the CPG regularization.

To confirm that the imputation accuracy of FIST is significantly improved by the CPG regularization, we showed in Figure 5.11 the performance of FIST in each of the 3 replicates by varying the graph hyper-parameter λ in the spot-wise evaluation. It is also consistent with the observation in the previous Figure 5.6, we can observe remarkable reduction on the MAE and MAPE and improvement on R^2 by increasing λ to 0.1. The

observation also verifies that the CPG topology is informative for the imputation task.

5.4 Discussions

In this chapter, we proposed to apply tensor modeling of multidimensional structure in spatially-resolved gene expression data mapped by the 2D spatial array. To the best of our knowledge, this is the first work to model the imputation of spatially-resolved transcriptomes as a tensor completion problem. Our key observations in the experiments with the ten 10x Genomics Visium spatial transcriptomic datasets are that 1) the imputation accuracy is significantly improved by leveraging the tensor representation of the sptRNA-seq data; 2) by incorporating the spatial graph and PPI network, the imputation accuracy and the content of the functional information in the imputed spatial gene expressions can be further improved significantly; and 3) FIST is capable of detecting gene clusters with more spatial characteristics that are consistent with the physiological features of the tissue.

Overall, we concluded that FIST is an effective and easy-to-use approach for reliable imputation of spatially-resolved gene expressions by modeling the spatial relation among the spots in the spatial array and the functional relation among the genes. The imputation results by FIST are both more accurate and functionally interpretable. FIST is also highly generalizable to other spatial transcriotomics datasets with high scalability and only one hyper-parameter needed to tune.

The current form of FIST solves the imputation problem within the 2D spatial array, and it becomes challenging to integrate multiple tissue samples (such as biological replicates) due to the difficulty of aligning the spatial grids across arrays. An interesting future direction is to formulate the imputation problem as a multi-task learning problem, such that the missing gene expression in multiple tissue samples can be imputed all together. To achieve this goal, we plan to model the multi-task learning problem as joint decomposition of multiple tensors to allow the information transfer across different tissue samples to enhance the imputation performance, based on the fact that the genes are shared across all the tissue samples.

Chapter 6

Theoretical Analysis

In this chapter, we provide theoretical analyses of the LowrankTLP, GT-COPR and FIST algorithms. In Section 6.1, we first show that LowrankTLP is based on a principled optimization framework in Equation (4.1) in Section 4.3.1 whose globally optimal solution minimizes an estimating error bound of recovering the true multi-relational tensor. Then, we provide a data-dependent *transductive Rademacher* bound to theoretically justify LowrankTLP for binary hyperlink prediction. In Sections 6.2 and 6.3, we adopt the similar procedures as in non-negative matrix factorization (NMF) [47] using the *auxiliary function* idea to prove the convergence of GT-COPR and FIST, as both algorithms use the multiplicative updating rule to minimize the non-convex objectives given in Equations (3.1) and (5.1), which are bounded from below by zero.

6.1 Error analysis of LowrankTLP algorithm

In this section, we first present an estimating error bound of LowrankTLP for multiple graph alignment, assuming the initial tensor $\mathcal{Y}^0$ is fully observed with Gaussian noise. The analysis provides a theoretical justification of the proposed optimization framework in Equation (4.1) in Section 4.3.1. Next, we use the *transductive Rademacher complexity* [65] to derive a data-dependent error bound of LowrankTLP for binary hyperlink prediction.

6.1.1 Estimating error bound for recovering TPG-structured tensor

Denote the transformation matrix $(1-\alpha)(I-\alpha S)^{-1}$ as P. We assume that the noisy tensor $\mathcal{Y}^0$ approximated by the pair-wise associations as described in Section 4.2, is generated with the true TPG-structured tensor $\mathcal{Y}^{true}$ and a noise tensor $\mathcal{Z} \in \mathbb{R}^{I_n \times I_{n-1} \times \cdots \times I_1}$ as

$$\mathbf{vec}(\mathcal{Y}^0) = P^{-1}\mathbf{vec}(\mathcal{Y}^{true}) + \mathbf{vec}(\mathcal{Z}),$$

where the entries of $\mathcal{Z}$ are drawn from the i.i.d Gaussian distribution $\mathcal{N}(0, \sigma^2)$.

Theorem 6.1.1. *Let $\hat{P} = (1-\alpha)(I-\alpha S_k)^{-1}$, where S_k is defined in Section 4.3.1 with $eig(S_k)$ selected from $eig(S)$ by Algorithm 4.1. The inferred tensor $\hat{\mathcal{Y}}^*$ found by the LowrankTLP algorithm as $\mathbf{vec}(\hat{\mathcal{Y}}^*) = \hat{P}\mathbf{vec}(\mathcal{Y}^0)$ in Equation* (4.4)*, has the following bounded recovery error to the true tensor $\mathcal{Y}^{true}$*

$$\mathbb{E}_{\mathcal{Z}}[||\hat{\mathcal{Y}}^* - \mathcal{Y}^{true}||_{\mathcal{F}}] \leq (1-\alpha)\left(\frac{\alpha|\lambda^*|}{1-\alpha\lambda^*}||\mathcal{Y}^0||_{\mathcal{F}} + \sigma\sqrt{\sum_{i=1}^{N}\frac{1}{(1-\alpha\lambda_i)^2}}\right), \tag{6.1}$$

where λ_is are the eigenvalues of S, λ^* is defined in Section 4.3.2 Equation (4.3)*, and $||.||_{\mathcal{F}}$ denotes the Frobenius norm of a tensor.*

Proof.

$$\begin{aligned}
&\mathbb{E}_{\mathcal{Z}}[||\hat{\mathcal{Y}}^* - \mathcal{Y}^{true}||_{\mathcal{F}}] = \mathbb{E}_{\mathcal{Z}}[||(\hat{P}-P)\mathbf{vec}(\mathcal{Y}^0) + P\mathbf{vec}(\mathcal{Z})||_2] \\
&\leq ||(\hat{P}-P)\mathbf{vec}(\mathcal{Y}^0)||_2 + \mathbb{E}_{\mathcal{Z}}[||P\mathbf{vec}(\mathcal{Z})||_2] \qquad (6.2)\\
&\leq ||(\hat{P}-P)\mathbf{vec}(\mathcal{Y}^0)||_2 + \sqrt{tr(\mathbb{E}_{\mathcal{Z}}[\mathbf{vec}(\mathcal{Z})\mathbf{vec}(\mathcal{Z})^T]P^TP)} \qquad (6.3)\\
&= ||(\hat{P}-P)\mathbf{vec}(\mathcal{Y}^0)||_2 + \sigma||P||_F \\
&\leq ||\hat{P}-P||_2||\mathbf{vec}(\mathcal{Y}^0)||_2 + \sigma||P||_F \\
&= (1-\alpha)\left(\frac{\alpha|\lambda^*|}{1-\alpha\lambda^*}||\mathcal{Y}^0||_{\mathcal{F}} + \sigma\sqrt{\sum_{i=1}^{N}\frac{1}{(1-\alpha\lambda_i)^2}}\right),
\end{aligned}$$

where Inequalities (6.2) and (6.3) are obtained with Minkowski's and Jensen's inequalities respectively (End of Proof). □

Since $\lambda_i \in [-1, 1], \forall i = 1, \ldots, N$ are constants for a fixed TPG, the second term on the right of Inequality (6.1) is approximated as $O(\sqrt{N})$; the upper bound of the

expected estimating error in Inequality (6.1) can be minimized by properly choosing λ^* to minimize $\frac{\alpha|\lambda^*|}{1-\alpha\lambda^*}$, which is the same as solving the optimization problem (4.1) in Section 4.3.1 as stated in Theorem 4.3.1. Thus, Theorem 6.1.1 theoretically justifies the proposed optimization formulation. Note that, the error bound in Inequality (6.1) reduces to its second term as

$$\mathbb{E}_{\mathcal{Z}}[||\mathcal{Y}^* - \mathcal{Y}^{true}||_{\mathcal{F}}] \leq (1-\alpha)\sigma\sqrt{\sum_{i=1}^{N}\frac{1}{(1-\alpha\lambda_i)^2}}, \tag{6.4}$$

when using the original TPG S without approximation, where $\mathcal{Y}^*$ is the closed-form solution such that $vec(\mathcal{Y}^*) = Pvec(\mathcal{Y}^0)$. Inequality (6.4) can be easily validated following the proof of Theorem 6.1.1.

As discussed so far, we proposed to approximate the transformation matrix P through properly selecting k eigen-pairs of S to minimize the estimating error bound in Theorem 6.1.1. Note that instead of using our approximation, another natural alternative is to directly find the best rank-k approximation to P. Proposition 6.1.1 below says that our approximation strategy is a better solution.

Proposition 6.1.1. *Follow the definitions of A and $\hat{A}$ in Proposition 4.3.1, where $eig(S_k)$ is selected from $eig(S)$ by Algorithm 4.1. Define A_k as the best rank-k approximation to A in both spectral and Frobenius norm. Assuming $k < \prod_i I_i$, we have the following inequalities*

$$||\hat{A} - A||_2 < ||A_k - A||_2 \text{ and } ||\hat{A} - A||_F < ||A_k - A||_F.$$

Proof. Let $\sigma_1 > \sigma_2 > \cdots > \sigma_N$ be the sorted eigenvalues of matrix S. Since $\sigma_i \in [-1, 1]$, for $i = 1, \ldots, N$ and $\alpha \in (0, 1)$, by Eckart-Young-Mirsky theorem, the non-zero eigenvalues of A_k are $\{\frac{1}{1-\alpha\sigma_i} : i = 1, \ldots, k\}$ and the perturbations are given as

$$||A_k - A||_2 = \frac{1}{1-\alpha\sigma_{k+1}} \text{ and}$$
$$||A_k - A||_F = \sqrt{\sum_{i=k+1}^{N}(\frac{1}{1-\alpha\sigma_i})^2}.$$

Using $\{\sigma_i : i = 1, \ldots, k\}$ as eigenvalues and their corresponding eigenvectors of S to construct a rank-k matrix L, and define $B = (I - \alpha L)^{-1}$, we have $||\hat{A} - A||_2 \leq ||B - A||_2$

and $||\hat{A} - A||_F \leq ||B - A||_F$ according to the definition of S_k in Section 4.3.1. Thus, inequalities in Proposition 6.1.1 hold if we can prove $||B - A||_2 < ||A_k - A||_2$ and $||B - A||_F < ||A_k - A||_F$. We first obtain the perturbations as

$$||B - A||_2 = \frac{\alpha|\sigma^*|}{1 - \alpha\sigma^*} \text{ and}$$
$$||B - A||_F = \sqrt{\sum_{i=k+1}^{N} (\frac{\alpha|\sigma_i|}{1 - \alpha\sigma_i})^2},$$

where $\sigma^* = \text{argmax}_{\sigma \in \{\sigma_{k+1}, \ldots, \sigma_N\}} \frac{\alpha|\sigma|}{1-\alpha\sigma}$. Now we need to show the Inequalities (6.5) and (6.6) are valid.

$$||B - A||_2 < ||A_k - A||_2 \tag{6.5}$$
$$||B - A||_F < ||A_k - A||_F \tag{6.6}$$

It is easy to prove Inequality (6.6) by the fact that $\alpha|\sigma_i| < 1$. To show Inequality (6.5), we have to consider three special cases: firstly, if $\sigma^* > 0$ and $\sigma_{k+1} \geq 0$ then we have $\sigma^* = \sigma_{k+1}$, thus Inequality (6.5) holds by the fact that $\alpha|\sigma^*| < 1$; secondly, if $\sigma^* < 0$ and $\sigma_{k+1} \geq 0$ we have $\frac{\alpha|\sigma^*|}{1-\alpha\sigma^*} < 1$ and $\frac{1}{1-\alpha\sigma_{k+1}} \geq 1$, thus Inequality (6.5) holds; finally, if $\sigma^* < 0$ and $\sigma_{k+1} < 0$ we have $|\sigma^*| \geq |\sigma_{k+1}|$, and $\frac{\alpha|\sigma^*|}{1-\alpha\sigma^*} < \frac{1}{1-\alpha\sigma^*} \leq \frac{1}{1-\alpha\sigma_{k+1}}$, thus Inequality (6.5) holds. Overall, we have shown

$$||\hat{A} - A||_2 \leq ||B - A||_2 < ||A_k - A||_2 \text{ and}$$
$$||\hat{A} - A||_F \leq ||B - A||_F < ||A_k - A||_F.$$

□

(End of Proof)

6.1.2 Transductive Rademacher bound for binary hyperlink prediction

Define $\Theta = \Theta_h \cup \bar{\Theta}_h = \{(i_1, i_2, \ldots, i_n) : \forall i_j \in [1, I_j], j = 1, \ldots, n\}$ as the set of all n-way associations among the nodes across the n knowledge graphs, where $\bar{\Theta}_h$ denotes the complement of Θ_h. Define tensor $\mathcal{Y}^{true} \in \{+1, -1\}^{I_n \times I_{n-1} \times \ldots \times I_1}$ which stores the true labels of all the hyperlinks in set Θ, where the label of the $(i_1, i_2, \ldots, i_n)$-th hyperlink

is either 1 (the link exists) or -1 (the link does not exist). Accordingly, $\mathcal{Y}^0$ contains a subset of known entries (hyperlinks) sampled from $\mathcal{Y}^{true}$ and zeros for the other unknown entries. Define $\mathcal{Y}_{out} \subset \mathbb{R}^{I_n \times I_{n-1} \times \ldots \times I_1}$ as the set of tensors outputted by the LowrankTLP algorithm over all possible Θ_h / $\bar{\Theta}_h$ partitions such that for every $\widehat{\mathcal{Y}}^* \in \mathcal{Y}_{out}$ we have $\mathbf{vec}(\hat{\mathcal{Y}}^*) = \hat{P}\mathbf{vec}(\mathcal{Y}^0)$ (as in Theorem 6.1.1). In the following derivations, we assume $\mathcal{Y}^0$ is normalized by $||\mathcal{Y}^0||_{\mathcal{F}}$ so that its Frobenius norm is unit. This normalization is proper since it does not change the signs in $\mathcal{Y}^*$. In Theorem 6.1.2, we provide a data-dependent error bound of LowrankTLP for binary hyperlink prediction, using the *transductive Rademacher complexity* proposed in [65].

Theorem 6.1.2. *Denote $l = |\Theta_h|$ and $u = |\bar{\Theta}_h|$ to be the cardinalities of Θ_h and $\bar{\Theta}_h$ respectively. Let $c_0 = \sqrt{\frac{32 \ln (4e)}{3}}$, $Q = \frac{1}{l} + \frac{1}{u}$ and $G = \frac{l+u}{(l+u-0.5)(1-0.5/\max(l,u))}$. For any fixed positive real γ, with probability of at least $1-\delta$ over the random choice of the set Θ_h, for all $\widehat{\mathcal{Y}}^* \in \mathcal{Y}_{out}$,*

$$\mathcal{L}_u^\gamma(\widehat{\mathcal{Y}}^*) \leq \widehat{\mathcal{L}}_l^\gamma(\widehat{\mathcal{Y}}^*) + \frac{||\hat{P}||_F}{\gamma}\sqrt{\frac{2}{lu}} + c_0 Q\sqrt{\min(l,u)} + \sqrt{\frac{GQ}{2}\ln(\frac{1}{\delta})}, \tag{6.7}$$

where $\mathcal{L}_u^\gamma(\widehat{\mathcal{Y}}^*)$ and $\widehat{\mathcal{L}}_l^\gamma(\widehat{\mathcal{Y}}^*)$ are the γ-margin test and empirical errors respectively defined as

$$\mathcal{L}_u^\gamma(\widehat{\mathcal{Y}}^*) = \frac{1}{u}\sum_{(i_1,i_2,\ldots,i_n)\in\bar{\Theta}_h} \ell_\gamma(\widehat{\mathcal{Y}}^*_{i_n,i_{n-1},\ldots,i_1}, \mathcal{Y}^{true}_{i_n,i_{n-1},\ldots,i_1})$$
$$\widehat{\mathcal{L}}_l^\gamma(\widehat{\mathcal{Y}}^*) = \frac{1}{l}\sum_{(i_1,i_2,\ldots,i_n)\in\Theta_h} \ell_\gamma(\widehat{\mathcal{Y}}^*_{i_n,i_{n-1},\ldots,i_1}, \mathcal{Y}^{true}_{i_n,i_{n-1},\ldots,i_1}),$$

with $\ell_\gamma(a,b) = 0$ if $ab > \gamma$ and $\ell_\gamma(a,b) = \min(1, 1-\frac{ab}{\gamma})$ otherwise.

Proof. The bound (6.7) in Theorem 6.1.2 is based on the transductive Rademacher bound using the *transductive Rademacher complexity* [65] given in Definition A.3.1 in appendix. According to Theorem A.3.1 in appendix, we only need to bound the

Rademacher complexity $R_{l+u}(\mathcal{Y}_{out})$ as below:

$$\begin{aligned}
R_{l+u}(\mathcal{Y}_{out}) &= (\frac{1}{l}+\frac{1}{u})\mathbb{E}_{\boldsymbol{\sigma}}\Big[\sup_{\widehat{\mathcal{Y}}^*\in\mathcal{Y}_{out}} \boldsymbol{\sigma}^T\mathbf{vec}(\widehat{\mathcal{Y}}^*)\Big] \\
&\leq (\frac{1}{l}+\frac{1}{u})\mathbb{E}_{\boldsymbol{\sigma}}\Big[\sup_{\mathcal{Y}^0:||\mathcal{Y}^0||_{\mathcal{F}}=1} \boldsymbol{\sigma}^T\hat{P}\mathbf{vec}(\mathcal{Y}^0)\Big] \\
&= (\frac{1}{l}+\frac{1}{u})\mathbb{E}_{\boldsymbol{\sigma}}\Big[||\hat{P}\boldsymbol{\sigma}||_2\Big] \qquad (6.8)\\
&\leq (\frac{1}{l}+\frac{1}{u})\sqrt{tr(\mathbb{E}_{\boldsymbol{\sigma}}\Big[\boldsymbol{\sigma}\boldsymbol{\sigma}^T\Big]\hat{P}^T\hat{P})} \qquad (6.9)\\
&= ||\hat{P}||_F\sqrt{\frac{2}{lu}},
\end{aligned}$$

where (6.8) and (6.9) are obtained using Cauchy-Schwarz and Jensen's inequalities respectively. (End of Proof) □

Given the eigenvalues of $\hat{P}$ are bounded within $[\frac{1-\alpha}{1+\alpha}, 1]$, it is clear that $||\hat{P}||_F \leq \sqrt{l+u}$. Assuming $l+u \to \infty$ and $l \ll u$, the error bound (6.7) can be simplified as $\mathcal{L}_u^{\gamma}(\widehat{\mathcal{Y}}^*) \leq \widehat{\mathcal{L}}_l^{\gamma}(\widehat{\mathcal{Y}}^*) + O\Big(\sqrt{\frac{1}{l}}\Big)$, which has a slower convergence rate compared with the estimating error bound of the convex tensor completion model [108] under certain conditions. It is also important to note that when l is very small i.e. the labeled n-way associations are extremely sparse, the term $\sqrt{\frac{1}{l}}$ increases relatively slow as l decreases. Thus, the bound by $O\Big(\sqrt{\frac{1}{l}}\Big)$ implies that empirically, the performance of LowrankTLP might deteriorate less with very sparse input tensors, which is consistent with our observations in both simulations and experiments on real datasets shown in Section 4.4.

6.2 Convergence analysis of GT-COPR algorithm

In this section, we analyze the convergence of GT-COPR with strong product graph regularization (given in Algorithm 3.1). The same analyses are also applicable to Cartesian and tensor product graph regularization with slight modifications.

As the objective function $\mathcal{J}$ in Equation (3.1) is clearly bounded from below by zero, the convergence of the updating rule given in Theorem 3.2.1 can be proved by showing that $\mathcal{J}$ is non-increasing under updating. We adopt the the *auxiliary function* defined in Theorem 6.2.1 to prove the convergence.

Theorem 6.2.1. *Lee and Seung [47]: A function $\mathcal{J}(h)$ is non-increasing under the update $h^* \leftarrow \arg\min_h G(h, \tilde{h})$ if $G(h, \tilde{h})$ is an auxiliary function for $\mathcal{J}(h)$, such that the following conditions are satisfied:*

$$G(h, \tilde{h}) \geq \mathcal{J}(h), \quad G(h, h) = \mathcal{J}(h).$$

By Theorem 6.2.1, the convergence claimed in Theorem 3.2.1 can be proved if the rule given in Theorem 3.2.1 is an update of one proper *auxiliary function* of $\mathcal{J}(A^{(i)})$, which is defined in Theorem 6.2.2.

Theorem 6.2.2. *The following function*

$$G(A_{ab}^{(i)}, \tilde{A}_{ab}^{(i)}) = \mathcal{J}(\tilde{A}_{ab}^{(i)}) + \mathcal{J}'(\tilde{A}_{ab}^{(i)})(A_{ab}^{(i)} - \tilde{A}_{ab}^{(i)}) + \frac{(X^4 \tilde{A}^{(i)} X^5 + X^6 \tilde{A}^{(i)} X^7 + \tilde{A}^{(i)} X^8 + \beta \tilde{A}^{(i)})_{ab}}{2\tilde{A}_{ab}^{(i)}} (A_{ab}^{(i)} - \tilde{A}_{ab}^{(i)})^2 \tag{6.10}$$

is an auxiliary function of $\mathcal{J}(A_{ab}^{(i)})$ and has its global minimum.

Proof: First, it is obvious that $G(A_{ab}^{(i)}, A_{ab}^{(i)}) = \mathcal{J}(A_{ab}^{(i)})$. To show $G(A_{ab}^{(i)}, \tilde{A}_{ab}^{(i)}) \geq \mathcal{J}(A_{ab}^{(i)})$ we obtain the second-order Taylor expansion of $\mathcal{J}(A_{ab}^{(i)})$ at the point $\tilde{A}_{ab}^{(i)}$ as

$$\mathcal{J}(A_{ab}^{(i)}) = \mathcal{J}(\tilde{A}_{ab}^{(i)}) + \mathcal{J}'(\tilde{A}_{ab}^{(i)})(A_{ab}^{(i)} - \tilde{A}_{ab}^{(i)}) + \frac{1}{2}\mathcal{J}''(\tilde{A}_{ab}^{(i)})(A_{ab}^{(i)} - \tilde{A}_{ab}^{(i)})^2,$$

with the second-order derivative given below:

$$\mathcal{J}''(\tilde{A}_{ab}^{(i)}) = -X_{aa}^2 X_{bb}^3 + X_{aa}^4 X_{bb}^5 + X_{aa}^6 X_{bb}^7 + X_{bb}^8 + \beta.$$

Thus, the Inequality $G(A_{ab}^{(i)}, \tilde{A}_{ab}^{(i)}) \geq \mathcal{J}(A_{ab}^{(i)})$ holds if

$$\frac{(X^4 \tilde{A}^{(i)} X^5 + X^6 \tilde{A}^{(i)} X^7 + \tilde{A}^{(i)} X^8 + \beta \tilde{A}^{(i)})_{ab}}{\tilde{A}_{ab}^{(i)}} \geq \mathcal{J}''(\tilde{A}_{ab}^{(i)}),$$

which can be demonstrated by the facts that $X_{aa}^2 X_{bb}^3 \geq 0$,

$$(X^4 \tilde{A}^{(i)} X^5)_{ab} = \sum_{l,m} X_{al}^4 \tilde{A}_{lm}^{(i)} X_{mb}^5 \geq X_{aa}^4 X_{bb}^5 \tilde{A}_{ab}^{(i)},$$

$$(X^6 \tilde{A}^{(i)} X^7)_{ab} = \sum_{l,m} X_{al}^6 \tilde{A}_{lm}^{(i)} X_{mb}^7 \geq X_{aa}^6 X_{bb}^7 \tilde{A}_{ab}^{(i)},$$

$$\text{and } (\tilde{A}^{(i)} X^8)_{ab} = \sum_{l} \tilde{A}_{al}^{(i)} X_{lb}^8 \geq X_{bb}^8 \tilde{A}_{ab}^{(i)}. \quad \text{(End of Proof)}$$

As the *auxiliary function* $G(A_{ab}^{(i)}, \tilde{A}_{ab}^{(i)})$ in Equation (6.11) is a quadratic function on variable $A_{ab}^{(i)}$, its minimum can be easily obtained in a closed-form as

$$\begin{aligned} A_{ab}^{(i)*} &= \arg\min_{A_{ab}^{(i)}} G(A_{ab}^{(i)}, \tilde{A}_{ab}^{(i)}) \\ &= \frac{(\tilde{A}^{(i)})_{ab}(X^1 + X^2\tilde{A}^{(i)}X^3)_{ab}}{(X^4\tilde{A}^{(i)}X^5 + X^6\tilde{A}^{(i)}X^7 + \tilde{A}^{(i)}X^8 + \beta\tilde{A}^{(i)})_{ab}}, \end{aligned}$$

which leads to the updating rule in Theorem 3.2.1.

To analyze the optimality of the fixed point after convergence, we first define $\{\Lambda^{(i)} \in \mathbb{R}^{I_i \times K} : i = 1, \ldots, n\}$ to be the matrices of Lagrange multipliers with the Lagrange function

$$\mathcal{L} = \mathcal{J} - \sum_{i=1}^{n} \mathrm{tr}(\Lambda^{(i)} A^{(i)T}).$$

Settin $\frac{\partial \mathcal{L}}{\partial A^{(i)}}$ to be zero, we obtain $\Lambda^{(i)} = \frac{\partial \mathcal{J}}{\partial A^{(i)}}$. Furthermore, when $A^{(i)}$ is a fixed point under the updating in Theorem 3.2.1 we have

$$(-X^1 - X^2A^{(i)}X^3 + X^4A^{(i)}X^5 + X^6A^{(i)}X^7 + A^{(i)}X^8 + \beta A^{(i)})_{ab}(A^{(i)})_{ab} = 0,$$

which implies the KKT complementary slackness condition $\Lambda_{ab}^{(i)} A_{ab}^{(i)} = 0$ is satisfied.

6.3 Convergence analysis of FIST algorithm

In this section, we show that FIST can converge under the updating rules in Equation (5.10)-(5.12), using the *auxiliary function* idea as in Section 6.2. Here, we only show that $\mathcal{J}$ is non-increasing under Equations (5.10). The proof is directly applicable to Equations (5.11) and (5.12). We first expand the derivative in Equation (5.5) as

$$\frac{\partial \mathcal{J}}{\partial \hat{A}_p} = -X_1 - \hat{A}_p X_2 - W_p \hat{A}_p X_3 + X_4 + \hat{A}_p X_5 + D_p \hat{A}_p X_3,$$

where $X_1 = (\mathcal{M}_{(1)} \circledast \mathcal{T}_{(1)})(\hat{A}_x \odot \hat{A}_y)$, $X_2 = \lambda(\Phi_x \circledast \Theta_y^W + \Phi_y \circledast \Theta_x^W)$, $X_3 = \lambda(\Phi_x \circledast \Phi_y)$, $X_4 = (\mathcal{M}_{(1)} \circledast \hat{\mathcal{T}}_{(1)})(\hat{A}_x \odot \hat{A}_y)$, and $X_5 = \lambda(\Phi_x \circledast \Theta_y^D + \Phi_y \circledast \Theta_x^D)$.

Based on Theorem 6.2.1, $\mathcal{J}$ is non-increasing under the update in Equation (5.10) if it is an update of one proper *auxiliary function* of $\mathcal{J}(\hat{A}_p)$, which is defined in Theorem 6.3.1.

Theorem 6.3.1. *The following function*

$$G([\hat{A}_p]_{a,b}, [\tilde{A}_p]_{a,b}) = \mathcal{J}([\tilde{A}_p]_{a,b}) + \mathcal{J}'([\tilde{A}_p]_{a,b})([\hat{A}_p - \tilde{A}_p]_{a,b}) + \frac{[X_4 + \tilde{A}_p X_5 + D_p \tilde{A}_p X_3]_{a,b}}{2[\tilde{A}_p]_{a,b}}([\hat{A}_p - \tilde{A}_p]_{a,b})^2 \tag{6.11}$$

is an auxiliary function of $\mathcal{J}([\hat{A}_p]_{a,b})$ *and has its global minimum.*

Proof. First, it is obvious that $G([\hat{A}_p]_{a,b}, [\hat{A}_p]_{a,b}) = \mathcal{J}([\hat{A}_p]_{a,b})$. To show $G([\hat{A}_p]_{a,b}, [\tilde{A}_p]_{a,b}) \geq \mathcal{J}([\hat{A}_p]_{a,b})$ we obtain the second-order Taylor expansion of $\mathcal{J}([\hat{A}_p]_{a,b})$ at the point $[\tilde{A}_p]_{a,b}$ as

$$\begin{aligned}\mathcal{J}([\hat{A}_p]_{a,b}) = \mathcal{J}([\tilde{A}_p]_{a,b}) + \mathcal{J}'([\tilde{A}_p]_{a,b})([\hat{A}_p]_{a,b} - [\tilde{A}_p]_{a,b}) \\ + \frac{1}{2}\mathcal{J}''([\tilde{A}_p]_{a,b})([\hat{A}_p]_{a,b} - [\tilde{A}_p]_{a,b})^2,\end{aligned}$$

with the second-order derivative given below:

$$\mathcal{J}''([\tilde{A}_p]_{a,b}) = -[X_2]_{b,b} - [W_p]_{a,a}[X_3]_{b,b} + [X_5]_{b,b} + [D_p]_{a,a}[X_3]_{b,b}$$

Thus, the Inequality $G([\hat{A}_p]_{a,b}, [\tilde{A}_p]_{a,b}) \geq \mathcal{J}([\hat{A}_p]_{a,b})$ holds if

$$\frac{[X_4 + \tilde{A}_p X_5 + D_p \tilde{A}_p X_3]_{a,b}}{[\tilde{A}_p]_{a,b}} \geq \mathcal{J}''([\tilde{A}_p]_{a,b}),$$

which can be demonstrated by the facts that

$$[D_p \tilde{A}_p X_3]_{a,b} = \sum_{l,m}[D_p]_{a,l}[\tilde{A}_p]_{l,m}[X_3]_{m,b} \geq [D_p]_{a,a}[X_3]_{b,b}[\tilde{A}_p]_{ab},$$

$$\text{and } [\tilde{A}_p X_5]_{a,b} = \sum_{l}[\tilde{A}_p]_{a,l}[X_5]_{l,b} \geq [X_5]_{b,b}[\tilde{A}_p]_{a,b}.$$

(End of Proof) □

As the *auxiliary function* $G([\hat{A}_p]_{a,b}, [\tilde{A}_p]_{a,b})$ in Equation (6.11) is a quadratic function on variable $[\hat{A}_p]_{a,b}$, its minimum can be easily obtained in a closed-form as

$$\begin{aligned}[\hat{A}_p]^*_{a,b} &= \arg\min_{[\hat{A}_p]_{a,b}} G([\hat{A}_p]_{a,b}, [\tilde{A}_p]_{a,b}) \\ &= \frac{[\tilde{A}_p]_{a,b}[X_1 + \tilde{A}_p X_2 + W_p \tilde{A}_p X_3]_{a,b}}{[X_4 + \tilde{A}_p X_5 + D_p \tilde{A}_p X_3]_{a,b}},\end{aligned}$$

which leads to the updating rule in Equation (5.10).

To analyze the optimality of the fixed point after convergence, we first define $\{\Lambda_p \in \mathbb{R}^{n_p \times r}, \Lambda_x \in \mathbb{R}^{n_x \times r}, \Lambda_y \in \mathbb{R}^{n_y \times r}\}$ to be the matrices of Lagrange multipliers with the Lagrange function

$$\mathcal{L} = \mathcal{J} - \sum_{i \in \{p,x,y\}} \mathrm{tr}(\Lambda_i \hat{A}_i^T).$$

Setting $\frac{\partial \mathcal{L}}{\partial \hat{A}_p}$ to be zero, we obtain $\Lambda_p = \frac{\partial \mathcal{J}}{\partial \hat{A}_p}$. Furthermore, when $A^{(i)}$ is a fixed point under the updating in Equation (5.10) we have

$$[-X_1 - \hat{A}_p X_2 - W_p \hat{A}_p X_3 + X_4 + \hat{A}_p X_5 + D_p \hat{A}_p X_3]_{a,b} [\hat{A}_p]_{a,b} = 0,$$

which implies the KKT complementary slackness condition $[\Lambda_p]_{a,b}[A_p]_{ab} = 0$ is satisfied.

Chapter 7

Conclusions

In this thesis, we approached the challenges in multi-relational learning as described in Section 1.4 from three aspects: 1) develop flexible and reliable tensor-based learning methods to support different forms of training associations; 2) develop theoretically justified estimation frameworks to enhance the efficiency and salability of the multi-relational learning methods; 3) adapt the tensor-based methods proposed in this thesis to infer the underlying multi-way associations in the data of emerging bioinformatics tasks. We showed that the tensor modeling and the product graph regularization made our methods more effective than existing relational learning methods to infer biologically important multi-way associations using flexible forms of training associations. We also showed that our methods, which rely on the optimal estimations of the tensor and product graph, were applicable to learn high-order multi-way associations across a large number of networks with high accuracy, and met the requirements of many bioinformatics applications. As follows, we summarize the contributions and technologies of the multi-relational learning methods proposed in this thesis.

In Chapter 3, we presented GT-COPR, the first method to directly learn a multi-relational tensor from the observed bipartite associations across multiple biological networks. GT-COPR regularizes the tensor with the graph Laplacian of a Cartesian, tensor or strong product graph, and constraints the consistencies between the collapsed tensors and the observed bipartite associations. We proved that GT-COPR significantly outperformed the existing methods which are based on matrix factorization/completion with the bipartite relational matrix, to predict the disease-gene-chemical associations

across the large-scale protein-protein interactions network, chemical structural similarity network and phenotype-based human disease network.

In Chapter 4, we presented LowrankTLP, the first method to generalize label propagation to the tensor product of a large number of graphs to learn a high-order multi-relational tensor. We proved that LowrankTLP, learns a subset of the eigen-pairs in the spectrum of the normalized tensor product graph (TPG) to minimize the upper bound of the noisy tensor estimating error, allowed the classical label propagation model to learn very high-order associations in a compressed tensor for the tasks of hyperlink prediction and multiple network alignment in broad applications. We also proved that the optimization framework in LowrankTLP, optimally estimates the global spectrum of TPG with respective to the learning objective, significantly improved the predictive accuracy compared to existing approximation approaches.

In Chapter 5, we presented FIST, the first method to model sptRNA-seq data as a tensor and formulate the imputation task as a tensor completion problem. We proved that FIST, explores the multi-way associations between genes and tissue locations, and gene functional associations carried in the PPI network, could overcome the high dropout rate of mRNAs in in-situ capture and complete the profiling of the gene expressions with higher accuracy than several best-performing methods that have been applied for the imputation of the scRNA-seq data. The tensor modeling also enabled FIST to capture the spatial characteristics in the gene expressions and reveal functions that are highly relevant to different tissue regions.

In summary, we conclude that all the methods proposed in this thesis benefited from the novel tensor modeling and efficient tensor computation, were more suitable for a wide range of multi-relational learning scenarios in bioinformatics than the traditional methods, and achieved state-of-the-art predictive performance.

References

[1] Björn H Junker and Falk Schreiber. *Analysis of biological networks*, volume 2. John Wiley & Sons, 2011.

[2] KH Young. Yeast two-hybrid: so many interactions,(in) so little time.... *Biology of reproduction*, 58(2):302–311, 1998.

[3] J Keith Joung, Elizabeth I Ramm, and Carl O Pabo. A bacterial two-hybrid selection system for studying protein–dna and protein–protein interactions. *Proceedings of the National Academy of Sciences*, 97(13):7382–7387, 2000.

[4] Tudor Groza, Sebastian Köhler, Dawid Moldenhauer, Nicole Vasilevsky, Gareth Baynam, Tomasz Zemojtel, Lynn Marie Schriml, Warren Alden Kibbe, Paul N Schofield, Tim Beck, et al. The human phenotype ontology: semantic unification of common and rare disease. *The American Journal of Human Genetics*, 97(1):111–124, 2015.

[5] Sebastian Köhler, Leigh Carmody, Nicole Vasilevsky, Julius O B Jacobsen, Daniel Danis, Jean-Philippe Gourdine, Michael Gargano, Nomi L Harris, Nicolas Matentzoglu, Julie A McMurry, et al. Expansion of the human phenotype ontology (hpo) knowledge base and resources. *Nucleic acids research*, 47(D1):D1018–D1027, 2019.

[6] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2019 update: improved access to chemical data. *Nucleic acids research*, 47(D1):D1102–D1109, 2019.

[7] Dinesh Kumar Barupal and Oliver Fiehn. Chemical similarity enrichment analysis (chemrich) as alternative to biochemical pathway mapping for metabolomic datasets. *Scientific reports*, 7(1):14567, 2017.

[8] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of cheminformatics*, 7(1):20, 2015.

[9] Francesco Iorio, Theo A Knijnenburg, Daniel J Vis, Graham R Bignell, Michael P Menden, Michael Schubert, Nanne Aben, Emanuel Gonçalves, Syd Barthorpe, Howard Lightfoot, et al. A landscape of pharmacogenomic interactions in cancer. *Cell*, 166(3):740–754, 2016.

[10] Chen Chen, Hanghang Tong, Lei Xie, Lei Ying, and Qing He. Fascinate: fast cross-layer dependency inference on multi-layered networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 765–774, 2016.

[11] Chih-Hsu Lin, Daniel M Konecki, Meng Liu, Stephen J Wilson, Huda Nassar, Angela D Wilkins, David F Gleich, and Olivier Lichtarge. Multimodal network diffusion predicts future disease–gene–chemical associations. *Bioinformatics*, 35(9):1536–1543, 2019.

[12] Chung-Shou Liao, Kanghao Lu, Michael Baym, Rohit Singh, and Bonnie Berger. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–i258, 2009.

[13] Vladimir Gligorijević, Noël Malod-Dognin, and Nataša Pržulj. Fuse: multiple network alignment via data fusion. *Bioinformatics*, 32(8):1195–1203, 2016.

[14] Rohit Singh, Jinbo Xu, and Bonnie Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, 105(35):12763–12768, 2008.

[15] Ferhat Alkan and Cesim Erten. Beams: backbone extraction and merge strategy for the global many-to-many alignment of multiple ppi networks. *Bioinformatics*, 30(4):531–539, 2014.

[16] Raphael Petegrosso, Sunho Park, Tae Hyun Hwang, and Rui Kuang. Transfer learning across ontologies for phenome–genome association prediction. *Bioinformatics*, 33(4):529–536, 2017.

[17] Maoqiang Xie, Taehyun Hwang, and Rui Kuang. Prioritizing disease genes by bi-random walk. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 292–303. Springer, 2012.

[18] Taehyun Hwang and Rui Kuang. A heterogeneous label propagation algorithm for disease gene discovery. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 583–594. SIAM, 2010.

[19] Emma Pierson and Christopher Yau. Zifa: Dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome biology*, 16(1):1–10, 2015.

[20] Sandhya Prabhakaran, Elham Azizi, Ambrose Carr, and Dana Pe'er. Dirichlet process mixture model for correcting technical variation in single-cell gene expression data. In *International Conference on Machine Learning*, pages 1070–1079, 2016.

[21] Davide Risso, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and Jean-Philippe Vert. A general and flexible method for signal extraction from single-cell rna-seq data. *Nature communications*, 9(1):1–17, 2018.

[22] Atsuhiro Narita, Kohei Hayashi, Ryota Tomioka, and Hisashi Kashima. Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery*, 25(2):298–324, 2012.

[23] Vincent W Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[24] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[25] Hisashi Kashima, Tsuyoshi Kato, Yoshihiro Yamanishi, Masashi Sugiyama, and Koji Tsuda. Link propagation: A fast semi-supervised learning algorithm for link

prediction. In *Proceedings of the 2009 SIAM international conference on data mining*, pages 1100–1111. SIAM, 2009.

[26] Rudy Raymond and Hisashi Kashima. Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs. *Machine Learning and Knowledge Discovery in Databases*, pages 131–147, 2010.

[27] Hanxiao Liu and Yiming Yang. Cross-graph learning of multi-relational associations. In *International Conference on Machine Learning*, pages 2235–2243, 2016.

[28] Denny Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004.

[29] Hiroki Sayama. Estimation of laplacian spectra of direct and strong product graphs. *Discrete Applied Mathematics*, 205:160–170, 2016.

[30] Andrew Chatr-Aryamontri, Rose Oughtred, Lorrie Boucher, Jennifer Rust, Christie Chang, Nadine K Kolas, Lara O'Donnell, Sara Oster, Chandra Theesfeld, Adnane Sellam, et al. The biogrid interaction database: 2017 update. *Nucleic acids research*, 45(D1):D369–D379, 2017.

[31] Feixiong Cheng, Chuang Liu, Jing Jiang, Weiqiang Lu, Weihua Li, Guixia Liu, Weixing Zhou, Jin Huang, and Yun Tang. Prediction of drug-target interactions and drug repositioning via network-based inference. *PLoS computational biology*, 8(5):e1002503, 2012.

[32] Hansaim Lim, Aleksandar Poleksic, Yuan Yao, Hanghang Tong, Di He, Luke Zhuang, Patrick Meng, and Lei Xie. Large-scale off-target identification using fast and accurate dual regularized one-class collaborative filtering and its application to drug repurposing. *PLoS computational biology*, 12(10):e1005135, 2016.

[33] Oron Vanunu and Roded Sharan. A propagation-based algorithm for inferring gene-disease associations. In *German Conference on Bioinformatics*. Gesellschaft für Informatik e. V., 2008.

[34] Chih-Hsu Lin, Daniel M Konecki, Meng Liu, Stephen J Wilson, Huda Nassar, Angela D Wilkins, David F Gleich, and Olivier Lichtarge. Multimodal network diffusion predicts future disease–gene–chemical associations. *Bioinformatics*, 2018.

[35] Martin Szummer and Tommi Jaakkola. Partially labeled classification with markov random walks. In *Advances in neural information processing systems*, pages 945–952, 2002.

[36] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University, 2002.

[37] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning*, pages 912–919, 2003.

[38] Deng Cai, Xiaofei He, Xiaoyun Wu, and Jiawei Han. Non-negative matrix factorization on manifold. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 63–72. IEEE, 2008.

[39] Quanquan Gu, Jie Zhou, and Chris Ding. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 199–210. SIAM, 2010.

[40] Yuan Yao, Hanghang Tong, Guo Yan, Feng Xu, Xiang Zhang, Boleslaw K Szymanski, and Jian Lu. Dual-regularized one-class collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 759–768. ACM, 2014.

[41] Atsuhiro Narita, Kohei Hayashi, Ryota Tomioka, and Hisashi Kashima. Tensor factorization using auxiliary information. *Data Mining and Knowledge Discovery*, 25(2):298–324, 2012.

[42] Evrim Acar, Daniel M Dunlavy, and Tamara G Kolda. Link prediction on evolving data using matrix and tensor factorizations. In *2009 IEEE International conference on data mining workshops*, pages 262–269. IEEE, 2009.

[43] Umang Sharan and Jennifer Neville. Temporal-relational classifiers for prediction in evolving domains. In *2008 Eighth IEEE International Conference on Data Mining*, pages 540–549. IEEE, 2008.

[44] Allan Peter Davis, Cynthia J Grondin, Robin J Johnson, Daniela Sciaky, Benjamin L King, Roy McMorran, Jolene Wiegers, Thomas C Wiegers, and Carolyn J Mattingly. The comparative toxicogenomics database: update 2017. *Nucleic acids research*, 45(D1):D972–D978, 2016.

[45] David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082, 2017.

[46] Anna Gaulton, Anne Hersey, Michał Nowotka, A Patrícia Bento, Jon Chambers, David Mendez, Prudence Mutowo, Francis Atkinson, Louisa J Bellis, Elena Cibrián-Uhalte, et al. The chembl database in 2017. *Nucleic acids research*, 45(D1):D945–D954, 2016.

[47] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[48] Wanjuan Yang, Jorge Soares, Patricia Greninger, Elena J Edelman, Howard Lightfoot, Simon Forbes, Nidhi Bindal, Dave Beare, James A Smith, I Richard Thompson, et al. Genomics of drug sensitivity in cancer (gdsc): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic acids research*, 41(D1):D955–D961, 2012.

[49] Chris Ding, Xiaofeng He, and Horst D Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 606–610. SIAM, 2005.

[50] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T

Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[51] Atanas Kamburov, Rachel Cavill, Timothy MD Ebbels, Ralf Herwig, and Hector C Keun. Integrated pathway-level analysis of transcriptomics and metabolomics data with impala. *Bioinformatics*, 27(20):2917–2918, 2011.

[52] David Y Zhang and Allen S Anderson. The sympathetic nervous system and heart failure. *Cardiology clinics*, 32(1):33–45, 2014.

[53] Markus Grube, Sabine Ameling, Michel Noutsias, Kathleen Köck, Ivonne Triebel, Karina Bonitz, Konrad Meissner, Gabriele Jedlitschky, Lars R Herda, Markus Reinthaler, et al. Selective regulation of cardiac organic cation transporter novel type 2 (octn2) in dilated cardiomyopathy. *The American journal of pathology*, 178(6):2547–2559, 2011.

[54] Jia Long, Chan-Juan Zhang, Neng Zhu, Ke Du, Yu-Fang Yin, Xi Tan, Duan-Fang Liao, and Li Qin. Lipid metabolism and carcinogenesis, cancer development. *American journal of cancer research*, 8(5):778, 2018.

[55] Viola Tamási, Katalin Monostory, Russell A Prough, and András Falus. Role of xenobiotic metabolism in cancer: involvement of transcriptional and mirna regulation of p450s. *Cellular and Molecular Life Sciences*, 68(7):1131–1146, 2011.

[56] Brett W. Bader, Tamara G. Kolda, et al. Matlab tensor toolbox version 2.6. Available online, February 2015.

[57] Jonathan Jiang. Multi-label learning on tensor product graph. In *AAAI*, 2012.

[58] Hanxiao Liu and Yiming Yang. Bipartite edge prediction via transductive learning over product graphs. In *International Conference on Machine Learning*, pages 1880–1888, 2015.

[59] Ruochen Xu, Yiming Yang, Hanxiao Liu, and Andrew Hsi. Cross-lingual text classification via model translation with limited dictionaries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 95–104. ACM, 2016.

[60] Xingwei Yang, Lakshman Prasad, and Longin Jan Latecki. Affinity learning with diffusion on tensor product graph. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):28–38, 2013.

[61] Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *IEEE proceeding*, 2015.

[62] Shaden Smith and George Karypis. SPLATT: The Surprisingly ParalleL spArse Tensor Toolkit. `http://cs.umn.edu/~splatt/`, 2016.

[63] Vladimir Gligorijević, Noël Malod-Dognin, and Nataša Pržulj. Fuse: multiple network alignment via data fusion. *Bioinformatics*, 32(8):1195–1203, 2015.

[64] Somaye Hashemifar, Qixing Huang, and Jinbo Xu. Joint alignment of multiple protein–protein interaction networks via convex optimization. *Journal of Computational Biology*, 23(11):903–911, 2016.

[65] Ran El-Yaniv and Dmitry Pechyony. Transductive rademacher complexity and its applications. *Journal of Artificial Intelligence Research*, 35:193–234, 2009.

[66] Quanquan Gu and Jiawei Han. Towards active learning on graphs: An error bound minimization approach. In *2012 IEEE 12th International Conference on Data Mining*, pages 882–887. IEEE, 2012.

[67] Annie Wang, Hansaim Lim, Shu-Yuan Cheng, and Lei Xie. Antenna, a multi-rank, multi-layered recommender system for inferring reliable drug-gene-disease associations: Repurposing diazoxide as a targeted anti-cancer therapy. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(6):1960–1967, 2018.

[68] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.

[69] Brett W Bader and Tamara G Kolda. Efficient matlab computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30(1):205–231, 2007.

[70] Shaden Smith, Niranjay Ravindran, Nicholas D Sidiropoulos, and George Karypis. Splatt: Efficient and parallel sparse tensor-matrix multiplication. *29th IEEE International Parallel & Distributed Processing Symposium*, 2015.

[71] Ferhat Alkan and Cesim Erten. Beams: backbone extraction and merge strategy for the global many-to-many alignment of multiple ppi networks. *Bioinformatics*, 30(4):531–539, 2013.

[72] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[73] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, 2011.

[74] Evrim Acar, Tamara G. Kolda, and Daniel M. Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. In *MLG'11: Proceedings of Mining and Learning with Graphs*, August 2011, 1105.3422.

[75] Ryota Tomioka, Kohei Hayashi, and Hisashi Kashima. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*, 2010.

[76] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998. ACM, 2008.

[77] Daniel Park, Rohit Singh, Michael Baym, Chung-Shou Liao, and Bonnie Berger. Isobase: a database of functionally related proteins across ppi networks. *Nucleic acids research*, 39(suppl_1):D295–D300, 2010.

[78] Gloria H Heppner. Tumor heterogeneity. *Cancer research*, 44(6):2259–2265, 1984.

[79] Felix Schmidt and Thomas Efferth. Tumor heterogeneity, single-cell sequencing, and drug resistance. *Pharmaceuticals*, 9(2):33, 2016.

[80] Diego Adhemar Jaitin, Ephraim Kenigsberg, Hadas Keren-Shaul, Naama Elefant, Franziska Paul, Irina Zaretsky, Alexander Mildner, Nadav Cohen, Steffen Jung, Amos Tanay, et al. Massively parallel single-cell rna-seq for marker-free decomposition of tissues into cell types. *Science*, 343(6172):776–779, 2014.

[81] Evan Z Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R Bialas, Nolan Kamitaki, Emily M Martersteck, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, 2015.

[82] Allon M Klein, Linas Mazutis, Ilke Akartuna, Naren Tallapragada, Adrian Veres, Victor Li, Leonid Peshkin, David A Weitz, and Marc W Kirschner. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, 161(5):1187–1201, 2015.

[83] Daniel Hebenstreit. Methods, challenges and potentials of single cell rna-seq. *Biology*, 1(3):658–667, 2012.

[84] Serena Liu and Cole Trapnell. Single-cell transcriptome sequencing: recent advances and remaining challenges. *F1000Research*, 5, 2016.

[85] Eric Lubeck, Ahmet F Coskun, Timur Zhiyentayev, Mubhij Ahmad, and Long Cai. Single-cell in situ rna profiling by sequential hybridization. *Nature methods*, 11(4):360, 2014.

[86] Je Hyuk Lee, Evan R Daugharthy, Jonathan Scheiman, Reza Kalhor, Joyce L Yang, Thomas C Ferrante, Richard Terry, Sauveur SF Jeanty, Chao Li, Ryoji Amamoto, et al. Highly multiplexed subcellular rna sequencing in situ. *Science*, 343(6177):1360–1363, 2014.

[87] Sheel Shah, Eric Lubeck, Maayan Schwarzkopf, Ting-Fang He, Alon Greenbaum, Chang Ho Sohn, Antti Lignell, Harry MT Choi, Viviana Gradinaru, Niles A Pierce, et al. Single-molecule rna detection at depth by hybridization chain reaction and tissue hydrogel embedding and clearing. *Development*, 143(15):2862–2867, 2016.

[88] Kok Hao Chen, Alistair N Boettiger, Jeffrey R Moffitt, Siyuan Wang, and Xiaowei Zhuang. Spatially resolved, highly multiplexed rna profiling in single cells. *Science*, 348(6233), 2015.

[89] Sanja Vickovic, Patrik L Ståhl, Fredrik Salmén, Sarantis Giatrellis, Jakub Orzechowski Westholm, Annelie Mollbrink, José Fernández Navarro, Joaquin Custodio,

Magda Bienko, Lesley-Ann Sutton, et al. Massive and parallel expression profiling using microarrayed single-cell sequencing. *Nature communications*, 7(1):1–9, 2016.

[90] Tal Nawy. Spatial transcriptomics. *Nature Methods*, 15(1):30–30, 2018.

[91] Sanja Vickovic, Gökcen Eraslan, Fredrik Salmén, Johanna Klughammer, Linnea Stenbeck, Denis Schapiro, Tarmo Äijö, Richard Bonneau, Ludvig Bergenstråhle, José Fernandéz Navarro, et al. High-definition spatial transcriptomics for in situ tissue profiling. *Nature methods*, 16(10):987–990, 2019.

[92] Samuel G Rodriques, Robert R Stickels, Aleksandrina Goeva, Carly A Martin, Evan Murray, Charles R Vanderburg, Joshua Welch, Linlin M Chen, Fei Chen, and Evan Z Macosko. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science*, 363(6434):1463–1467, 2019.

[93] Antti Lignell, Laura Kerosuo, Sebastian J Streichan, Long Cai, and Marianne E Bronner. Identification of a neural crest stem cell niche by spatial genomic analysis. *Nature communications*, 8(1):1–11, 2017.

[94] Stefania Giacomello, Fredrik Salmén, Barbara K Terebieniec, Sanja Vickovic, Jose Fernandez Navarro, Andrey Alexeyenko, Johan Reimegård, Lauren S McKee, Chanaka Mannapperuma, Vincent Bulone, et al. Spatially resolved transcriptome profiling in model plant species. *Nature Plants*, 3(6):17061, 2017.

[95] Emelie Berglund, Jonas Maaskola, Niklas Schultz, Stefanie Friedrich, Maja Marklund, Joseph Bergenstråhle, Firas Tarish, Anna Tanoglidi, Sanja Vickovic, Ludvig Larsson, et al. Spatial maps of prostate cancer transcriptomes reveal an unexplored landscape of heterogeneity. *Nature communications*, 9(1):1–13, 2018.

[96] Eric A Smith and H Courtney Hodges. The spatial and genomic hierarchy of tumor ecosystems revealed by single-cell technologies. *Trends in cancer*, 5(7):411–425, 2019.

[97] Shao-Bo Liang and Li-Wu Fu. Application of single-cell technology in cancer research. *Biotechnology advances*, 35(4):443–449, 2017.

[98] Silas Maniatis, Tarmo Äijö, Sanja Vickovic, Catherine Braine, Kristy Kang, Annelie Mollbrink, Delphine Fagegaltier, Žaneta Andrusivová, Sami Saarenpää, Gonzalo Saiz-Castro, et al. Spatiotemporal dynamics of molecular pathology in amyotrophic lateral sclerosis. *Science*, 364(6435):89–93, 2019.

[99] Catalina A Vallejos, Davide Risso, Antonio Scialdone, Sandrine Dudoit, and John C Marioni. Normalizing single-cell rna sequencing data: challenges and opportunities. *Nature methods*, 14(6):565, 2017.

[100] Michaela Asp, Joseph Bergenstråhle, and Joakim Lundeberg. Spatially resolved transcriptomes — next generation tools for tissue exploration. *BioEssays*, page 1900221, 2020.

[101] Oliver Stegle, Sarah A Teichmann, and John C Marioni. Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 16(3):133–145, 2015.

[102] TaeHyun Hwang, Ze Tian, Rui Kuang, and Jean-Pierre Kocher. Learning on weighted hypergraphs to integrate protein interactions and gene expressions for cancer outcome prediction. In *2008 Eighth IEEE International Conference on Data Mining*, pages 293–302. IEEE, 2008.

[103] Ze Tian, TaeHyun Hwang, and Rui Kuang. A hypergraph-based learning algorithm for classifying gene expression and arraycgh data with prior knowledge. *Bioinformatics*, 25(21):2831–2838, 2009.

[104] Hyunsoo Kim and Haesun Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.

[105] Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. Blast+: architecture and applications. *BMC bioinformatics*, 10(1):421, 2009.

[106] Chen Xu and Zhengchang Su. Identification of cell types from single-cell transcriptomes using a novel clustering method. *Bioinformatics*, 31(12):1974–1980, 2015.

[107] Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33(5):495–502, 2015.

[108] Ryota Tomioka, Taiji Suzuki, Kohei Hayashi, and Hisashi Kashima. Statistical performance of convex tensor decomposition. In *Advances in neural information processing systems*, pages 972–980, 2011.

[109] Guangchuang Yu, Li-Gen Wang, Yanyan Han, and Qing-Yu He. clusterprofiler: an r package for comparing biological themes among gene clusters. *Omics: a journal of integrative biology*, 16(5):284–287, 2012.

[110] Steven D Crowley, Susan B Gurley, Michael I Oliverio, A Kathy Pazmino, Robert Griffiths, Patrick J Flannery, Robert F Spurney, Hyung-Suk Kim, Oliver Smithies, Thu H Le, et al. Distinct roles for the kidney and systemic tissues in blood pressure regulation by the renin-angiotensin system. *The Journal of clinical investigation*, 115(4):1092–1099, 2005.

[111] Thomas M Coffman and Steven D Crowley. Kidney in hypertension: guyton redux. *Hypertension*, 51(4):811–816, 2008.

[112] Sophia N Verouti, Emilie Boscardin, Edith Hummler, and Simona Frateschi. Regulation of blood pressure and renal function by ncc and enac: lessons from genetically engineered mice. *Current opinion in pharmacology*, 21:60–72, 2015.

[113] Dennis Brown and Carsten A Wagner. Molecular mechanisms of acid-base sensing by the kidney. *Journal of the American Society of Nephrology*, 23(5):774–780, 2012.

[114] Haruko Yanase, Kumiko Takebe, Junko Nio-Kobayashi, Hiromi Takahashi-Iwanaga, and Toshihiko Iwanaga. Cellular expression of a sodium-dependent monocarboxylate transporter (slc5a8) and the mct family in the mouse kidney. *Histochemistry and cell biology*, 130(5):957–966, 2008.

[115] Shushi Nagamori, Pattama Wiriyasermkul, Meritxell Espino Guarch, Hirohisa Okuyama, Saya Nakagomi, Kenjiro Tadagaki, Yumiko Nishinaka, Susanna Bodoy,

Kazuaki Takafuji, Suguru Okuda, et al. Novel cystine transporter in renal proximal tubule identified as a missing partner of cystinuria-related plasma membrane protein rbat/slc3a1. *Proceedings of the National Academy of Sciences*, 113(3):775–780, 2016.

[116] Rudolfs K Zalups. Organic anion transport and action of γ-glutamyl transpeptidase in kidney linked mechanistically to renal tubular uptake of inorganic mercury. *Toxicology and applied pharmacology*, 132(2):289–298, 1995.

[117] Naohiko Anzai, Promsuk Jutabha, Atsushi Enomoto, Hirokazu Yokoyama, Hiroshi Nonoguchi, Taku Hirata, Katsuko Shiraya, Xin He, Seok Ho Cha, Michio Takeda, et al. Functional characterization of rat organic anion transporter 5 (slc22a19) at the apical membrane of renal proximal tubules. *Journal of Pharmacology and Experimental Therapeutics*, 315(2):534–544, 2005.

[118] AKIHIRO Tojo, TAKASHI Sekine, NORIKO Nakajima, MAKOTO Hosoyamada, YOSHIKATSU Kanai, KENJIRO Kimura, and HITOSHI Endou. Immunohistochemical localization of multispecific renal organic anion transporter 1 in rat kidney. *Journal of the American Society of Nephrology*, 10(3):464–471, 1999.

[119] Jin-Sun Hwang, Eun-Young Park, WJ Kim, Chul-Woo Yang, and Jin Kim. Expression of oat1 and oat3 in differentiating proximal tubules of the mouse kidney. *Histology and histopathology*, 2010.

[120] Patrik L Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O Westholm, Mikael Huss, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82, 2016.

[121] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Appendix A

Definitions and Lemmas

A.1 Basic tensor decomposition models

Definition A.1.1. ***Canonical polyadic decomposition (CPD):***
An n-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n}$ of rank r can be written as

$$\mathcal{X} = \sum_{c=1}^{r} \boldsymbol{a}_c^{(1)} \circ \boldsymbol{a}_c^{(2)} \circ \cdots \circ \boldsymbol{a}_c^{(n)} = [\![A^{(1)}, A^{(2)}, \ldots, A^{(n)}]\!],$$

where $\boldsymbol{a}_c^{(i)}$ is the c-th column of component matrix $A^{(i)} \in \mathbb{R}^{I_i \times r}$.
Vectorization property: *The vectorization of CPD-form is $vec(\mathcal{X}) = (A^{(n)} \odot A^{(n-1)} \odot \cdots \odot A^{(1)})\mathbf{1}$, where $\mathbf{1}$ is a vector with all-ones.*

Definition A.1.2. ***Tucker decomposition:***
An n-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n}$ can be decomposed into a core tensor $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times \ldots \times r_n}$ and component matrices $\{A^{(i)} \in \mathbb{R}^{I_i \times r_i} : i = 1, \ldots, n\}$ as

$$\mathcal{X} = \mathcal{G} \times_1 A^{(1)} \times_2 A^{(2)} \cdots \times_n A^{(n)} = [\![\mathcal{G}; A^{(1)}, A^{(2)}, \ldots, A^{(n)}]\!].$$

Vectorization property: *The vectorization of $\mathcal{X}$ is $vec(\mathcal{X}) = (A^{(n)} \otimes A^{(n-1)} \ldots \otimes A^{(1)}) vec(\mathcal{G})$.*

A.2 Some useful lemmas

Lemma A.2.1. *If A, B, C and D are matrices of such size that one can form the matrix products AC and BD, then $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$.*

Lemma A.2.2. *If matrices A, B, C and D are of such size that one can form the operation $(A \odot B), (C \odot D), (A^T C)$ and $(B^T D)$, then equality $(A \odot B)^T (C \odot D) = (A^T C) \circledast (B^T D)$ holds.*

Lemma A.2.3. *Let $\lambda_1, \ldots, \lambda_n$ be eigenvalues of A with corresponding eigenvectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, and let $\mu_1, \ldots, \mu_m$ be eigenvalues of B with corresponding eigenvectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$. Then the eigenvalues and eigenvectors of $A \otimes B$ are $\lambda_i \mu_j$ and $\boldsymbol{x}_i \otimes \boldsymbol{y}_j$, $i = 1, \ldots, n$, $j = 1, \ldots, m$.*

Lemma A.2.4. *Let matrix $\tilde{W}^{(i)}$ denote the best rank-k_i approximation to $W^{(i)}$ per Eckart-Young-Mirsky theorem [121]. The matrix $\otimes_{i=1}^{n} \tilde{W}^{(i)}$ is not guaranteed to be the best rank $\prod_{i=1}^{n} k_i$ approximation to $\otimes_{i=1}^{n} W^{(i)}$.*

A.3 Transductive Rademacher complexity

The *transductive Rademacher complexity* and the data-dependent error bound for binary transductive learning proposed in [65] are given below in Definition A.3.1 and Theorem A.3.1.

Definition A.3.1. *Given a fixed set $\Phi_{l+u} = \{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, l+u\}$ of sample-label pairs drown from an unknown distribution, w.l.o.g., the training set sampled uniformly without replacement from Φ_{l+u} is denoted as $\Phi_l = \{(\boldsymbol{x}_i, y_i) : i = 1, \ldots, l\}$, and the test set is denoted as $X_u = \{\boldsymbol{x}_i : i = l+1, \ldots, l+u\}$. Define $\mathcal{H}_{out} \subseteq \mathbb{R}^{l+u}$ as a set of vectors $\boldsymbol{h} = (h(\boldsymbol{x}_1), \ldots, h(\boldsymbol{x}_{l+u}))^T$ output by a transductive algorithm using the set Φ_l and X_u over all possible training/test set partitions, such that $h(\boldsymbol{x}_i)$ is the soft label of example $\boldsymbol{x}_i$. The* transductive Rademacher complexity *is defined as*

$$R_{l+u}(\mathcal{H}_{out}) = (\frac{1}{l} + \frac{1}{u}) \mathbb{E}_{\boldsymbol{\sigma}} \Big[\sup_{\boldsymbol{h} \in \mathcal{H}_{out}} \boldsymbol{\sigma}^T \boldsymbol{h} \Big],$$

where $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{l+u})^T$ is a vector of i.i.d random variables such that $\sigma_i = 1$ with probability p, $\sigma_i = -1$ with probability p and $\sigma_i = 0$ with probability $1 - 2p$. We set $p = \frac{lu}{(l+u)^2}$ as in [65].

Theorem A.3.1. *Let $c_0 = \sqrt{\frac{32 \ln (4e)}{3}}$, $Q = \frac{1}{l} + \frac{1}{u}$ and $G = \frac{l+u}{(l+u-0.5)(1-0.5/\max(l,u))}$. For any fixed positive real γ, with probability of at least $1 - \delta$ over the random training/test*

set partitioning, $\forall \boldsymbol{h} \in \mathcal{H}_{out}$,

$$\mathcal{L}_u^{\gamma}(\boldsymbol{h}) \leq \widehat{\mathcal{L}}_l^{\gamma}(\boldsymbol{h}) + \frac{R_{l+u}(\mathcal{H}_{out})}{\gamma} + c_0 Q \sqrt{\min(l,u)} + \sqrt{\frac{GQ}{2}\ln(\frac{1}{\delta})},$$

where $\widehat{\mathcal{L}}_l^{\gamma}(\boldsymbol{h}) = \frac{1}{l}\sum_{i=1}^{l} \ell_{\gamma}(h(\boldsymbol{x}_i), y_i)$ *and* $\mathcal{L}_u^{\gamma}(\boldsymbol{h}) = \frac{1}{u}\sum_{i=l+1}^{l+u} \ell_{\gamma}(h(\boldsymbol{x}_i), y_i)$ *are the* γ*-margin empirical and test error respectively, with* $\ell_{\gamma}(a,b) = 0$ *if* $ab > \gamma$ *and* $\ell_{\gamma}(a,b) = \min(1, 1 - \frac{ab}{\gamma})$ *otherwise.*